

The U.K. ATARI Computer Owners Club Issue 11 Price £1.00

Independent User Group

# Monitor

Look inside for  
Hex Code Generator  
6502 Machine Code  
BASIC Programs  
RAM Talker  
ET Bar

Lotus Reviews

Korons Rift

Goonies

Mercenary

Sidewinder

Electraglide

Alternate Reality

Fighter Pilot  
Review





*presents*



The **THIRD GENERATION** of Jeff Minter's unique **LIGHT SYNTHESISERS**  
- **MIND/MACHINE SYNERGY** -

**COLOURSPACE:** allows the user to create dynamic, interactive light displays using the Atari's 320x200 512-colour graphics. Do it to music... your ears won't believe your eyes...

**COLOURSPACE:** mouse control, over 100 keyboard commands, compatible with NEOchrome images (for foreground and background displays), 20 presets, 84 definable lightforms, record mode, load and save of performance data from disk.

**COLOURSPACE:** raw 68000 power harnessed to create a completely new artform... the first true light-synthesiser... play light like music...

**COLOURSPACE.** Discover why you bought your ST.

NOW AVAILABLE at £19.95, from retailers of ATARI software or direct from  
LLAMASOFT 49 Mount Pleasant Tedley Harts

Tel: 07356 4476

SEND S.A.E. FOR CATALOGUE & NEWSLETTER

## Atari Show

The first ever British Atari show is to be held at the Novotel, Hammersmith, London from Friday the 7th to Sunday the 9th of March 1986. All your favourite suppliers will be there, as will Atari themselves. The show is being organised by Database Publications who, as you probably know, publish *Atari User* magazine. This show has been a very long time in coming for us dedicated Atari enthusiasts, so lets make it a big success by getting along there and giving it good support. In that way it could turn into an annual event. And that would be great.

The fortunes of the Atari computer in this country have certainly taken a turn for the better in the first part of the year. Due to the move to the bargain prices of 8088, prices in the high street shops. Lets hope it continues right through the year and the Atari computer takes its rightful place as the best value home micro on the market. The ST is having a good time and is gaining in prestige with every day that passes. I am sure if you asked practically anybody what computer they would like to own, the answer would be the 586ST (or maybe the 1040ST now that we know its coming). But it takes time for the good news to spread sometimes, there are still many places, especially computer shops where you can get away with programs you like for a Commodore or Commodore 64, but if you ask for Atari, you are looked on with disdain and given a pointer. Sorry, I don't get asked for that stuff! Well the tide is turning, there are more and more Atari owners every day, so get in there and ask and keep asking until they are the light and shade the goodies. And they will if they are there as a market for the software. Many software houses who have ignored Atari in the past are seriously putting their foot in the water, they seem to be waking awake, its about time that they did too!

We get many requests from members for reviews of new programs. So in this issue of Monitor we present a bumper list of new titles that are coming soon or available now. All are excellent quality and as the prices have stabilised to about £10 for cassettes and £15 for disks, are all good value for money. Who said there was nothing new for Atari owners?

# CONTENTS

## 2 What's HOT?

Part 3 concludes with some MCM programs

## 6 RAM Talker

Listen to your own voice or your favorite singer

## 8 Book Reviews

Some of the best reading around

## 10 ST Music Pattern

Demo programs in ST Base

## 12 AtariWriter Plus

A snail's pace of this improved program from Atari Corp.

## 14 Starting from Basics

Part 4 of our series for beginners

## 17 Software Library

The quarter's new programs

## 18 Reviews

Software licenses including Sidewinder, Korosoft, Eternagade, Heremany, Fighter Pilot, Gears and Alternate Reality

## 24 Cooking the Code

Part 6 of our 6502 tutorial

## 28 Competition Result

Find out who was in the picture and what he was saying

## 29 Hexadecimal Code Generator

Useful utility giving a better presentation of your programs

## 30 8088L Disabler

Load battery files without holding down Option

## 30 Hot Glossy

Waspert on the graphics!

Don't be a lack of spirit, Opening Out and the Advertising Column have kept on!

## CREDITS

Editor	Ray Smith
Art Editor	Peter Blackmore
Technical Editor	Ron Levy
Technical Editor	Ruth Mayhew
Advertising Editor	Steve Hillen

## SUBSCRIPTION RATES

UK and Ireland	£6.00
Europe	£7.00
Outside Europe (Quarterly)	£10.00
Outside Europe (Annual)	£36.00

The above prices are in English Pounds

Sending and multiple postage and packing  
A subscription membership fee to join the U.K. Atari Computer Owners Club is just £1.00 for four issues of the club magazine, All

cheques, postal orders and to be made payable to the U.K. Atari Computer Owners Club. Overseas that membership is also available at slightly higher rates. Quarterly members who use the library service will not include through a subscription, or a relevant postage

## ADVERTISEMENTS

Please note that the club is not to be held legally responsible for items made by advertisers.

Cover: Korosoft Kit from Actionstar and Fighter Pilot from Digital Integration.  
Club Address: The U.K. Atari Computer Owners Club, P.O. Box 2, Rayleigh, Essex.

Copyright © 1986, ATARI COMPUTER OWNERS CLUB. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage and retrieval system, without permission in writing from the Atari Computer Owners Club. Printed in Great Britain by the Atari Computer Owners Club. All other names are trademarks of their respective owners.

# WHAT'S MIDI?

by Michael Stringer Part Three

In the second part of this short series I gave a brief review of the current musical instruments that were equipped with MIDI and the enormous potential the ATARI 520ST has as a Master Controller. Also covered in some detail were the complexities of creating new sounds from scratch and the storage of the data in a form of library. The MIDI language, the concept of MIDI WORDS and SENTENCES and the various categories were introduced.

The MIDI categories still to be covered are System Real Time, System Exclusive, System Common and System Reset. These topics will only be briefly described because of the overall complexity of the subject. In many respects it is comparable to using a computer — its internal architecture is very complex, but it is a simple device to actually use. The degree of ease is dependent upon the software. Similarly, to the end-user, MIDI consists of a pair of wires in its simplest form, but it can be controlled by a computer using suitable software.

## System Real Time

The requirements for transmitting live from a keyboard to a computer. In this instance, the computer is behaving like a rather sophisticated tape-recorder. The software requirements would produce some form of feedback in the form of a metronome or a couple of bars. Some form of accurate clock to synchronise everything is also required.

The MIDI Standard for timing is covered in this section. The clock is nominally 24 clock beats per eighth (Quarter note) with 96 per bar. This is covered in "Timing Clock in Play" (248). At the end of a bar or measure, "Timing Clock with Measure End" (249), is sent. "Start from First Measure" is transmitted

Status	Description
10110000	Timing Clock in Play
10110001	Timing Clock with Measure End
10110100	Start at First Measure
10110101	Continue
10111000	Timing Clock in Stop
10111100	System Reset

Table 1

Status	Following Bytes	Description
11110000	01111111	Reserved for special system functions.
	"	"
	"	MSB = 0, any number of bytes.
	11110111	End of block

Table 2

Status	Following Bytes	Description
10110000	00000111	Status byte
	00000111	Identification Number ( Yamaha? )
	00000111	Sub status and Channel number
	00000111	Parameter group number
	00000111	Parameter number
	00000111	Data
11110111		End of Exchange

Table 3

when it is intended to re-play the part. The System insists that the "First Timing Clock in Play" code is transmitted within 5ms. When it is intended to "Continue" (251) again the system insists that the "First Timing Clock in Play" code is transmitted within 5ms. When the system is stopped, or pausing, another code "Timing Clock in Stop" (252) is transmitted to re-synchronise everything. See Table 1.

## System Exclusive

This is the portion of the MIDI Standard that lays down the Status Bytes reserved for manufacturers to

design unique code for their own devices. See Table 2.

These codes have the highest priority within the system, with the exception of the System Reset. The Yamaha DKT System Exclusive is very extensive and by way of an example, I will describe only one, small, section. Under the heading of Parameter Change, the sequence of data bytes shown in Table 3 is transmitted.

If we now take Table 4, it is quite easy to see how, using just a few bytes, it is possible to control the enormous range of the principal parameters. Under the "d" column, which appears in the Odd/even (Data) section in Table 3, the degree of range is covered.

The p allocation 01 to 125 repeats the data for Operations 5 4 3 2 & 1.

This completes the formulae of the MIDI Standard. Although the coverage has been brief, I hope that it has been reasonably thorough. Throughout this discussion I have made it clear that the subject is not easy to grasp, especially if you do not have access to a MIDI equipped instrument, but if you do intend, at some time, to get involved in computer music generated



#	Parameter	#
0	Op. B.B. Note 1	0 = 99
1	" " 2	"
2	" " 3	"
3	" " 4	"
4	" Level 1	"
5	" " 2	"
6	" " 3	"
7	" " 4	"
8	Op. K.L.S. Break Point	"
9	" Left Depth	"
10	" Right Depth	"
11	" Left Curve	0 = 3
12	" Right Curve	"
13	Op. K.B. Note Scaling	0 = 7
14	Op. Modulation Sensitivity Step	0 = 3
15	Op. Key Velocity Sensitivity	0 = 7
16	Op. Operator Output Level	0 = 99
17	Op. Oscillator Mode	0 = 1
18	Op. Oscillator Frequency	0 = 31
19	" Fine	0 = 99
20	" Detune	0 = 14

# = 0	Parameter	#
121	Flash EQ Note 1	0 = 99
122	" " 2	"
123	" " 3	"
124	" " 4	"
125	" Level 1	"
126	" " 2	"
127	" " 3	"
128	" " 4	"
129	Algorithm Select	0 = 31
130	Parameter	0 = 7
131	Gen. Marker Synchronisation	0 = 1
132	L.F.D. Speed	0 = 99
133	" Delay	"
134	" P.F.D.	"
135	" A.F.D.	"
136	" Synchronisation	"
137	" Move	0 = 4
138	Modulations Flash Serial Velocity	0 = 7
139	Parameters	0 = 99
140	Waveform Character 1	ARC11
141	" 2	"
142	" 3	"
143	" 4	"
144	" 5	"
145	" 6	"
146	" 7	"
147	" 8	"
148	" 9	"
149	" 10	"
150	Operator Output	0 = 99

# = 2	Parameter	0 = 99	0 = 12	0 = 3	0 = 99	0 = 7	0 = 99	0 = 3	0 = 99	0 = 7
44	Temp / Poly Mode Change									
45	Flash Band Range									
46	" Step									
47	Portamento Note									
48	" 0.1 seconds									
49	" Time									
50	Modulation Wheel Range									
51	" Assign									
52	Foot Controller Range									
53	" Assign									
54	Breath Controller Range									
55	" Assign									
56	Aftertouch Range									
57	" Assign									

with the help of a 6205T and a synthesiser. I hope you will find this information will be of some use.

There have been some very interesting developments on the MIDI front with CASIO coming into the market with some very cheap, but quite respectable, instruments. One should also look at the possibility of picking up instruments second hand. The cover all quality is so good these days that this is not the risk that it was a few years ago. If the creation of exciting sounds, instead of actually playing appeals to you, consider the desktop known as expensiv. These little boxes contain the guts of a synthesiser and operate under MIDI. The output goes straight to an amplifier with the input and parameter settings under the control of a Master Unit. They are effectively synthesisers without a keyboard. The savings are quite considerable, but with no loss of quality. The cheapest will cost a few hundred pounds. The best, in my opinion, is the Yamaha TS615. This device consists of eight CS37s in one box. The sound quality has to be heard to be appreciated. The cost of eight CS37s is about £12,000 and the cost of the TS615 about £3,000. This will give you an indication of the sort of reductions that are possible. The best advice that I can give you is to pop into the local emporium, browse, ask questions and most importantly, listen. I am very fortunate in having two very good retail outlets in my vicinity (East) - "Monkey Torch" and "Monkey Business" (there are others, but I have not used them out. Due to the enormous current popularity of electronic music you will not have too far to look for similar shops.

On the ST front there have been a number of very interesting happenings. Since I wrote Part 2, BASIC has evolved and a number of "free-base". Two word processors and a Data Base to name but a few. Actually I am trying out 1ST WORD with the article I wrote in, I must admit, somewhat of a disappointment. It has some very interesting features, but this is so slow. Some examples of Basic programs appear shortly.

The most exciting event to report is an afternoon spent on the Hard disc. I must admit, in all honesty, that I don't expect to see it until the Summer, but it is here now in January! My first impressions are that it is outstanding. It is very fast. Some programs were up and running on the screen before my finger was off the mouse button! The capacity of this particular model was just over 20 Gb Mega Bytes. When I got down to storing sounds and scores in the library this is one device that I must have. Probably by the time you read this you will also have had the opportunity to try one out.

At this end there are two MIDI

program being developed. The first is a Sound and Score Editor, with a library for the DDT. This is, in effect, a 64 track tape recorder with full ranging facilities and a 32 Voice Score Editor with at least a 50,000 note capacity. This program will allow Step-time and also Real-time editing and storage for the user plus sound creating facilities and storage.

The second program is a MIDI Keyboard Tutor. This will follow the similar programs associated with Type writer Tutor. The emphasis will be on finger training through practice to develop touch and dexterity. Then, as soon as possible, to play starting at a Beginner's level to an intermediate level. The computer will not be too critical at first, but the criterion will increase as the pupil's experience and knowledge develops. The emphasis at the stage will be on note recognition and confidence, with play and follow and starts achieving this aim. At a later stage an Advanced level Tutor will be considered. The trouble is, it all takes a great deal of time to develop!

If you have a MIDI Keyboard, or access to one, I will conclude with three very simple programs in ST BASIC. The first, Listing 1, is a data receiver. Connect the keyboard to the MIDI ports on the ST via suitable cables, as shown in Part 1 for these programs. When you switch on, data from the synthesiser appears on the screen. The first string of data you will see is - 25. These are Status Bytes associated with timing - the 11111110 Active Sensing Bytes. These are continuously output in the rate of 1 per 80ms except during the transmission of bulk data, dumps and reception. If you now play a note you will notice the command for "KEY ON", "NOTE" and "KEY FORCE". If your keyboard doesn't have keyboard sensitivity this last piece of data is defaulted to 64. On releasing the note "KEY OFF", "NOTE" and "KEY FORCE" is transmitted. In this instance, the "KEY FORCE" is 0, effectively switching the note off. If you experiment with function controllers, such as PITCH BEND or MODULATION you will notice an enormous amount of data being transmitted. You can now appreciate why I mentioned in Part 1 that these devices are "memory gobblers".

The first thing to note about these data strings is that each value is returned as a minus (-) value. After a short period of experimentation the timing status data and Note On become a distraction. To turn these off simply add 25 IF A=-113 THEN 30 IF A=-2 THEN 20. A=A+255

It is important to realise that data coming from the synthesiser is always returned as a minus value - data that is

#### LISTING 1 MIDI DATA MONITOR

```
10 MIDI DATA MONITOR FOR THE SOUND
20 CLEAR:CLS:GOTO 3:PAUSE 2
30 A=INP(3)
50 W$=DOTO 20
```

```
25 IF A=-113 THEN 20 IF A=-2 THEN 20 A=A+255
```

#### Listing 2

#### LISTING 2 RANDOM CHORD GENERATOR

```
10 MIDI RANDOM CHORD GENERATOR
20 A=INT(113*RND(1)):OUT 3,254:OUT 3,0A:OUT 3,44
30 OUT 3,1A+4:OUT 3,44:OUT 3,1A+7:OUT 3,44
40 OUT 3,1A+12:OUT 3,44
50 FOR I=1 TO 300:NEXT I
60 OUT 3,1A:OUT 3,44:OUT 3,1A+4:OUT 3,44:OUT 3,1A+7:OUT 3,44
70 OUT 3,1A+12:OUT 3,44:GOTO 20
80 OUT 3,-25:GOTO 10
```

#### Listing 3

#### LISTING 3 A MIDI REAL TIME RECORDER

```
10 MIDI REAL TIME RECORDER
20 AUTHOR (2000000)
30 CLEAR:GOTO 100
40 IF INP(1)=1 THEN A=INP(3) ELSE GOTO 20
50 OUT 3,A-128:REMARK (20000,1)
60 A=0
70 GOTO 100:PRINT:GOTO 100:PLAY NOW, HIT PITCH BEND TO STOP
80 IF INP(1)=1 THEN F=1:GOTO 80
90 A=INP(3)+255
100 IF F=0 AND A < 144 THEN F=1:GOTO 80
110 REMARK (A,80)
120 MID$(A$+1,1)=A
130 F=0:GOTO 100
140 IF A=254 THEN 180
150 GOTO 80
160 TIME=121
170 GOTO 180
180 GOTO 15,18:PRINT:PLAYBACK
190 GOTO 100,12,12:PRINT:TEMP 24 TO 24:PRINT
200 FOR B=0 TO 9
210 IF REMARK(18,1)=0 THEN 230 ELSE FOR T=0 TO 100:PRINT
220 OUT 3,MID$(A$+1,1)
230 NEXT
240 GOTO 20
```

#### Listing 3

transmitted to the synthesiser is always a positive value. The simple program in Listing 2 is a random chord generator.

A is a random number generated to provide the root for the chord. To this number a 4th, 7th and 12th semi tone is added to construct the chord. Out 3 64 is the default value for the velocity which these notes are played. As each note is generated it is played and then each note is turned off. The FOR - NEXT loop at line 25 provides the duration that the chord is being played. The root value is printed on the screen for reference only.

The final program, Listing 3, I picked up somewhere. I do not know who the author is, or even if it is complete. I typed it in as an example and it

appears to work quite satisfactorily. It is a recorder of sequences that are re-played through a TEMPO control. It is very important to match the graphic buffer off before the program is run. Type the program in and save it, switch the buffer off, answer the prompt and then reload the program which you can then RUN. Failure to do this will result in an "OUT OF MEMORY" prompt. I know this sounds strange - but I did say earlier that I was not impressed with this particular version of BASIC!

Well, I hope you will have some fun with these little programs. Don't expect too much, the limitations are set by the speed of BASIC. Machine code routines would solve all of the problems, but that is another story.

**Don't miss the BIG  
Atari event of 1986!**

# ATARI COMPUTER SHOW



For the first time *Atari UK* and all the major suppliers of *Atari* hardware and software are pulling out all the stops to make the first-ever *Atari Computer Show* the top event of the year.

Everyone who's anyone in the *Atari* world will be there. Already many third-party suppliers are planning to use the show as a launching pad for products still on the drawing board.

If you're a long-standing *Atari* user the show will bring you right up to date on all the exciting developments now taking place in the ever-expanding *Atari* world.

And if you're one of the many thousands of newcomers to *Atari* computing it will open your eyes to the vast selection of *Atari* hardware and software that is now available for the whole *Atari* range.

**It's a show you cannot afford to miss!**

**Champagne Suite, Novotel,  
Hammersmith, London W6**



*'It's about time there was an Atari Computer Show. It's a major breakthrough in the world of Atari' - Jerry Howell, Managing Director, Software Express.*

*'It's good news about the show. It will develop a lot of interest among software producers' - Phillip Morris, Software Manager, English Software.*

*'We will give the show our total support. It is an exciting development in a fast moving market place' - Tony Deane, Marketing Director, Silica Shop.*

## **Exhibitors who have already booked stands:**

Atari UK  
2 Bit Systems  
Artic Publications  
Atari User  
Buildon ITEC  
Bee  
Boot-Out  
Cashlink  
CDS Software  
Compuart

Computer Bookshops  
Computer Concepts  
Datacube  
Direct Disc  
Escon  
GST  
Gulftronic  
Habe  
Interplayc Robots  
KECM

Kuma  
Llamasoft  
Megamax  
Microdel  
Micromet  
Mikes Computer Software  
Miniature Tool Co  
Page 6  
Prospero  
Ramco Electronics (UK)

Sleam Chip  
Slice Shop  
Software Express  
Software International  
Software Plus  
Solidix  
Sunshine Books  
Zonefour

# RAM TALKER

For 400 and 500 Machines Only

by Randy Holmes.

Regarded from STATUS,  
Norfolk, VA, U.S.A.

This program allows you to experiment with digital speech on your Atari micro. It requires a special circuit to allow the computer to read an analogue voltage at Port 2. Figure 1 shows the small circuit you will need to build. In place of the microphone, you may want to substitute a simple 1/4 inch jack socket to enable you to plug in a guitar, keyboard or tape player. This will give higher quality sound than recording from the mic. Adjust the volume control on your sound source to get the best distortion free results.

The components consist of a 0.1  $\mu$ F capacitor, a NPN transistor 2N2222 (this is just a general purpose type, almost any NPN transistor should work), a 100k $\Omega$  resistor and a 9-way 'D' type socket for connection to the joystick port.

You could build the circuit on a piece of veroboard and wire to pins 7 and 9 on the port. To test the circuit RUN the Ramtalker program and select the 'Throughput' option with a sample speed of 1.

Plug in a microphone, guitar, tape recorder or any other form of sound source, and see if any sound comes from your TV/Monitor speaker. If not, recheck your wiring, making sure all connections are good.

The program, Listing 1, is a friendly, fast, easy to use program. When typing it in, if not using NIBO (or some other checksum program like TYPO) then ignore the two code letters before each line number.

When the program is run, a menu is presented. To perform a desired function, press the number corresponding to the function. A BASIC GET command eliminates the need to press (RETURN) after selection.

If the function you pressed was not the one you wanted, pressing (RETURN) will take you back to the main menu. If you choose 'Record', you will be asked for a sample speed. This is the speed at which the program will read the information coming in at the port. A sample speed of 1 will render the highest quality sound, while a value of 255 will result in nearly unintelligible noise.

Once a sample speed is specified, followed by a (RETURN), press the (START) key to begin recording.

The 'Play' asks for a sample speed too. This will be the speed at which the

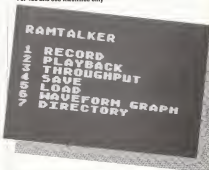


Figure 1.

sound information contained in memory will be played back. A good speed is usually around 10, giving a natural sound, but you may wish to detect the sound by giving it a higher or lower value. Again, pressing (START) after giving the sample speed will begin the playback.

'Throughput' asks for a recording sample speed, and will allow you to play sounds through the speaker with no time limitations. Press (START) to begin and (SYSTEM RESET) to get you out of the one.

'Save' and 'Load' ask you for a file name, include 'D', or 'C' in the file specification. The program uses the Central Input/Output (CIO) routines.







# BOOK REVIEWS

## The Easy Guide To Your ATARI 800XL/800XL

by Thomas Blackader  
Published by SAGEX UK  
Price £7.95

If you are an absolute beginner to the world of computing, then this is the book, or should it be said, the handbook for you. To help you understand and appreciate your 800XL/800XL computer it should also be noted that other than slight changes in the keyboard and the graphics availability, this book can also be applied to a 400 or 800 computer equally well. As this book was written primarily for the American market, reference is made to a 1300XL computer.

The American author has covered all the very basic points of computing in a very clear and concise language. Other than a reference to dollar prices, the book is very easy to understand and gives the reader every confidence to try and experiment with the information set out so explicitly.

The first section covers the actual setting up of the computer and peripherals, explaining the various keyboard functions and its control of the screen. Graphics is another subject very easy to understand. The section on writing programs, also covers use of variables as well as controlling your program, which encourages the reader to develop computer skills without having to wade through pages of technical jargon, which to the beginner is



often both confusing and misleading. Optional exercises are also set out in this section to give the beginner the opportunity to experiment with small programs and thus, hopefully, reap advanced ones.

Advanced graphics covers the control of the colours of the screen. Storage is discussed in the last section of the book and covers use of the Program Recorder and Disk Drive, with particular



emphasis on care of the disks. This does not only apply to beginners!

The commercial software referred to is all available in the UK.

An excellent and informative book, which is a must for all beginners who need an easy guide to enjoying and understanding the ATARI 800XL/800XL.

## Atari 130XE Machine Language for the Absolute Beginner

by Kevin Bergen  
Published by Melbourne House  
Price £7.95

With the recent marked increase in the ownership of ATARI computers over the Christmas period, two books have been published by Melbourne House enabling the beginner to develop computer skills, as well as, more importantly, enjoy the superb quality of the ATARI computer.

Although both books have been written for the 130XE, they are both compatible with the 800XL.

The first book by Kevin Bergen on Machine Language is, as the title indicates, written for the absolute beginner. I found it well written in such an easy to understand language making



the reader begin to feel confident and, therefore, experiment and develop the skills required in computing.

Every experienced programmer appreciates the importance of Machine Language, but as a beginner it is always very hard to get away from playing games which have been written commercially, and make the transition towards programming.

The 12 chapters of the book range from an introduction to Machine Language in chapter 1 to Program Control, Counting, Looping and Pointing as well as Table Storage. Where applicable, an exercise and summary is included after each chapter.

At the end of the book there is an appendix section which lists instruction codes as well as microprocessor operation codes and many other very useful codes and listings.

All in all a very useful and informative book.

# BOOK REVIEWS

## Atari 130XE Games Book

by Richard Woodcock  
and Gernie Strutton  
Published by Melbourne House  
Price £6.95

Although several books have been written for the ATARI, some have been good and some much less so. However, I feel that any new book which is nicely presented, easy to understand and very helpful, will find a place amongst my other computer books.

It is all too easy for the experienced computer programmer to disregard what we novices need to help and encourage us. In any new book, even if only one section helps the beginner it must be good.

This book (also for the 800XL) in fact, does both, it reminds the beginner that the Letters 'C' and 'T' cannot be used instead of Figures 10 and 11! Typing errors are important.

The book includes a CHECKSUM

program for validation — this small addition may save the beginner a great deal of time trying to find the error.

30 suggested programs are contained in the book, which depict various aspects of actual games; a logic skill, evasion strategy, Program variables and structure are set out as well as the actual listings. A small screen display is also included.

Each listing is self-headed so it becomes apparent which particular part of the program does what and the effect it has on the finished program. For a game like 'PENGUINUT' the headings are — Base on screen, Main Loop, Adjust score & serve, Game End, Drive Coast and then Initiate routine. This is set out clearly and is easy to understand.

Other programming modifications are also suggested after each section such as adding music, speeding up muscles to make the games more interesting.

I read this book very much, not too much data for me to input.



## The Robot Book

by Richard Pawson  
Published by Windward £7.95

If you have ever wondered what to do with all that Lego that has accumulated over the last few years and has now been stored away in the loft, then this is the book for you and your family.

The Robot Book is not only an enjoyable book to read, written in an entertaining way, but also educates the reader on how Robots come into existence and their subsequent progression into the high technological world of which we have all now become a part.

The first part covers the history of Computers/Robots from as early as the mid 1700's to the present day, where micro technology and artificial limbs, are still being developed.

Robots in action is covered extensively in Part (2) such as toys, floor robots as well as pseudo robots, which are becoming increasingly popular for shop-window advertising in major stores as well as on television e.g. the famous Johnnie Chase robotic advert for a well known television company.<sup>1</sup>

How Robots work is detailed very explicitly in Part (3), types of construction, programming, motive power, sensed and vision, as well as the use of computer interfaces are all covered in great detail.

Twelve projects are listed in this book, four of which use Lego. The



emphasis being on the beginner, with his basic approach Richard Pawson encourages electronically minded computer enthusiasts to try developing one of these robots. Step by step instructions are included with each project together with the relevant computer program.

Although none are for Atari computers it would probably be possible to adapt them. However, the basic construction can also be achieved equally well without high level knowledge of maths, electronics or advanced computer programming skills. Each section is very well illustrated with drawings and diagrams as well as some most attractive photographs.

With the increase in the number of young people studying Electronics/Computer Sciences at school, where pupils are encouraged to design their own Robots for domestic as well as industrial use, then this is a very useful addition to any school or home library. In all, this is an excellent book, and would make an ideal gift for the enthusiastic hobbyist as well as the budding computer/engineer!

# ST High-Res Pattern

by Josephine Glasgow

Here is an ST version of the popular 'hat' shape 3-D drawing lesson to many an eight-bit Atari owner.

Program 1 will produce a picture of about one quarter the size of the screen, using the output window (window 2), and it will perform hidden-line removal.

This means that, if you imagine the picture as a three dimensional, non-transparent object, you are only able to see the edges facing towards you. Program 2 on the other hand, produces a full-screen picture, but does not remove the hidden lines, in other words you are able to see right through the hat!

Program 1 works by simply plotting pixels at given intervals, so the lines formed can be either 'dotted' or in places, and experimenting with the FOR-NEXT loops on lines 160 and 190 will usually give less tidy results. Program 2 however, works by drawing lines between points, and so the resulting lines are more 'complete'. You can change the length of each line down each level by changing the STEP value

on line 190, the larger the value, the less curved the finished lines will be, but the faster it will run. Changing the STEP value in line 160 will (as with program 1) change the number curves down, so you could try experimenting with this as well.

Incidentally, the pictures shown have been produced by dumping the screen to an Epson printer. For those of

you who have not yet been 'bored' enough to seek out and buy an ST yet, a screen image is dumped to paper simply by holding the 'ALTERNATE' key down and pressing the 'HELP' key. The ST then stops whatever it is doing (regardless of what application or language is running!) and automatically prints a copy of whatever high-res image is on the screen. Very useful indeed!

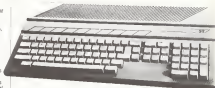


Figure 1



Figure 2

Program 1

```

10  FULLN 2
20  CLEARN 2
150 SP=144:SR=4.71238:SP=SR/SP
160 FOR ZI=-64 TO 64
170  ZT=ZI*2.25:ZT=ZT*ZT
180  SL=INT(SQR(28736-ZT))+0.5
190  FOR XI=0-SL TO SL
200   ST=SQR(SI+SI+ZS)*SF
210   YY=180W(XT)+SIN(XT+3)+0.4)*56
220   SI=XI+ZT+160:YI=YB-YY+31
230   COLOR 1,1,0,1
240   LINEF XI,YI,XI,YI
250   COLOR 1,0,0,0:LINEF XI,YI+1,XI,400
260 NEXT XI:NEXT ZI
    
```

Program 2

```

10  FULLN 2
20  CLEARN 2
150 SP=144:SR=4.71238:SF=SR/SP
160 FOR ZI=-64 TO 64 STEP 1
170  ZT=ZI*2.25:ZT=ZT*ZT
180  SL=INT(SQR(28736-ZT))+0.5
190  FOR XI=0-SL TO SL STEP 2
200   ST=SQR(SI+SI+ZS)*SF
210   YY=180W(XT)+SIN(XT+3)+0.4)*56
220   SI=XI+ZT+160:YI=YB-YY+31+1
230   IF SI=0 THEN X2=XI:Y2=YI
240   LINEF XI,YI,XI,YI
250 NEXT XI:ZT=0: NEXT ZI
300 GOTO 300
    
```

micro

# ATARI ST

MicroTron

**LANDS OF HAVOC** From Microdeal  
By Steve Bell, Microdeal's 2000+  
current arcade adventure with its full  
colour maps, the Book of Chicago and  
Satan, our reptile hero, has just been  
adapted to the first Mega-game  
available for the machine. Superb full  
colour graphics, moving with  
sound and written in 2000+  
machine code.  
Excellent value at ONLY

**£19.95**

Will run on  
both colour  
 & monochrome  
 systems



**MUDPES** From Microdeal  
Amold grabs a MUD PIE and  
throws it at the nearest clown, hitting  
him smack in the middle of the face!  
The clown, who didn't think it too  
funny, starts THROWING things back  
and it is your turn to NOW THE PUM  
SLAP! for MUDPES. Specially conceived  
things arcade game. Colour Monitor  
required. ONLY

**£19.95**

**FLIP SIDE** From Microdeal

This REVISED type game combines Goo-  
gle's death with Chicken like code for an  
excitement all of its own! Survived an  
opportunity price and FLIP is over. This  
simple rule leads to strategies as detailed as  
your imagination. For 1, 2 or 3 players, 1  
level of difficulty.

A Special option and a Help facility take  
new players through everything.

Superbly presented and

**ONLY £19.95**

Will run on both  
colour & monochrome  
systems



**M-DISK** From Microdeal A RAM - Disk emulator uses a  
portion of your computers memory to load, store and save  
data. IT ACTS LIKE A NORMAL DISK-DRIVE  
BUT has no mechanical parts. M-DISK is  
software based and can utilise up to 200K of  
memory (if it is available). M-DISK is super fast,  
incredibly tough, and costs

**ONLY £12.95**

**SOFT SPOOL**  
From Microdeal

While printing files (from

a word processor, TOS, etc.) your  
computer is simply dead weight, tied up  
waiting for the printer to finish. However,  
with SOFT SPOOL, the wait is over!  
NOW YOUR COMPUTER CAN PRINT  
AND THINK AT THE SAME TIME!

This software spools prints printed  
data to the printer before it starts  
freeing the Computer for more  
important things (like playing  
Mudoles maybe?) and saves  
you literally hours of computer time! An excellent  
utility and

**ONLY £12.95**

**SOFT SPOOL/M-DISK  
TWIN PACK**

Now the previously mentioned utilities available in 1 money saving  
Twin Pack. Contains both manuals plus 1 disk.

**ONLY £19.95**

**TIME BANDIT** By Jim Dunaway & Harry Laifner From Microdeal  
Battle Evil Guardians as you try to escape with the Treasures of  
Time. Each player leads every one else plus simultaneously, each  
on his own mission!

Use the Timegates to visit over 20 unique adventures scenes  
with over 15 levels each! Profound medieval dungeons, twisted  
monsters, and future worlds all in one game!

Unique features and 3-D terrain for each Time, beautiful  
scrolling landscapes, detailed animation, thrilling sound and  
music, hundreds of screens, the conquest of Time and Space  
awaits you!

For the Atari ST with color monitor **£29.95** Due out  
March/April Feb 1988

**M-COPY** By Timothy Patten From Microdeal  
Designed for duplicating program discs, M-Copy takes care of  
the time to make copies and ensures that the operator will notice  
when a copy fails.

After the source disk is copied into memory just once, all the  
operator has to do is put in blank discs.

M-COPY speeds through disk duplication because it formats  
and copies only those bytes needed by the program instead of  
wasting time on the rest of the disk. M-Copy moves on to the  
next disk. For the Atari ST a must for companies copying discs  
in bulk.

**£49.95**

Microdeal programs are available from  
ATARI ST Centers or by post.

Post & Packing 75p per order.

**MICROPOST**  
By Steve Bell, Microdeal

41 Tress Road, St Austell Cornwall PL26 5JE England  
Telephone: 0726 69020 Telex: 40816 MICROD G

**UTILITIES** By Timothy Patten From Microdeal

Keep control over your Atari ST files, even when things go  
wrong. With an ability to relocate individual bytes, you can  
change the order, change file and volume names, change file  
attributes, format individual disk tracks, copy individual sectors,  
restore deleted files, recover data from damaged disks and repair  
damaged disks.

Utilities work with the DOS interface to provide easy, menu-  
driven operations. Changing file contents, attributes or names is  
just a matter of typing in the new information or clicking on  
selector buttons.

Even restoring deleted files only requires that you input  
data, then click on a button to save or reject it. Sector-by-sector  
copy and track-by-track format routines let you gently handle  
damaged disks.

For the Atari ST **£39.95**

**DISK HELP** By Steve Bell From Microdeal

If your disk has crashed and you're not an experienced  
programmer able to repair it with UTILITIES then this easy to  
use menu-driven program will attempt to repair the disk and  
bring back lost data. This program will format a disk without  
erasing it and reverse the directory, check a disk and list bad  
sectors, repair them with bad sectors so they become readable  
even as physically damaged disks. If you are data free on your  
ST you cannot afford to be without DISK HELP. Can be used  
with 1 or 2 single or double sided drives.

**£29.95**

ATARI ST Centers or by post



# ATARIWRITER PLUS

A sneak preview.

By Jack Holzhauer

Reprinted from Current Notes.

Atariwriter Plus announced at summer 85 CES show in Las Vegas has been a long time in coming. Has it been worth the wait? Probably, although it may fall short of some people's expectations. How does it differ from the current version? Functionally both products are almost the same. If you are familiar with Atariwriter, using Plus will be a doddle. Major differences? For 130XE owners, memory capacity has been increased to 46 KB, enough for about 32 double-spaced pages.

The first thing you notice about Plus is that it is disk based. The program disk is double-sided; the standard version on one side and the XE version on the other. A second disk is also supplied containing a Proofreader Dictionary and a 66 page manual is included in the package.

## Main Menu & I/O Operations

The main menu screen will be familiar to all Atariwriter users, but two new functions allow you to load the Proofreader (and Mail Merge) modules. You can also get a directory listing on both drive 1 and drive 2. And guess what? The Windows on those multiple disks of yours don't scroll off the top of the screen whilst you're trying to read them. The first 36 lines are listed in a double column format. The rest are accessed with a tap on the Return key. At the bottom of the screen is a submenu allowing you to load, delete or save a file whilst the directory is right there in front of you on the screen.

Disk format operations can be directed to either drive 1 or 2 (enhanced density is forced if the system is booted with a 1020). By the way, all I/O commands now accept wild cards.

The only other new feature on the menu screen is the global format function. When accessed it shows all the default formatting settings, i.e. top margin, type font, justification, etc. You know all those funny symbols you're used to seeing on the top line of the edit screen. You can make any desired changes there and there.

When first accessing the PRINT command from the menu, you are allowed to choose printer drivers from any one of seven Atari models (825, 1025, 1027, 1020, 1029, and the new XM44, 801, and XM44 221). The Epson 1060, 1026 Monoprint 430, Jet 6-200 or a custom driver created with the printer driver module (see below). You can also allow it to print to devices other than your printer, to either drive or to RS232 ports H1 through to H4.



## Edit Screen

Pretty much the same, but there are a couple of very helpful new features. Plus allows you to format free edit screens up to 249 columns wide. Using the Insert? Set your columns to 96 Condensed? Set them to 132. Of course you can only view 40 columns, so you advance to the right the text scrolls off the left of the screen. Even so, this is a boon to anyone who has tried to columnise text using the current version. You can also switch from Insert to Typewrite mode for making corrections.

## Cursor Movement

Only one change here. Plus allows you to advance or backspace by word, but the two-fingered two-handed key stroke required (Select/Lt key/Right Arrow) is retained.

## Block Operations

Three new functions. Plus allows you to alphabetise a list of words or phrases. This might be useful in some situations. You can also obtain a word count on your entire text or any portion thereof. In addition you are allowed to save a designated block of text. Other block functions such as delete, move and duplicate are essentially unchanged. The buffer in the XE versions approximates 6K; in the non-XE version to the remaining free bytes.

## Search and Replace

About the same, but keyboard sequences are different. You may search for a control character, use ? as a wild card in your search string, search upward from the current cursor position, and search without having to respond to a replace query.

## Type Fonts and Embedded Printer Control Codes

The current version of Atariwriter provides the user with 8 default type fonts: Pica, Condensed & Proportional accessible through the G command. Plus adds 3 more: Elite, Superscript and Selectscript, the latter two claimed to be available only with Atari's new XM580 printer (they produce condensed superscript on my Epson compatible Panasonic 1022, just as intended). Both superscript and subscript are still available using the Select Up/Down Arrow sequences. Actually, Plus allows you to set up as many as eight default fonts using its printer driver utility. An 'emphasized' or 'bold' command is provided, and you can access the international character on your printer, if it has one. As in the original version, embedded printer control codes are accepted.

## Print Formatting

All the standard print formatting features remain unchanged: centering text, headers, justification, line spacing, page numbering, embedded section heading, etc., but Plus adds one additional feature, software supported double column printing. As far as I am concerned, this feature is worth the cost of upgrading! And both columns are presented side by side on the preview screen.

## Memory Management (130XE)

The 130XE version of Plus can handle 46K plus and achieves this by bank-switching three blocks of text each

handling US-872 bytes (the amount shown) first, when you first enter the edit screen. As you approach the capacity of the first bank of 128K, you must issue a command to leave the screen to the second bank. The switch is not done automatically, although saving from and loading to all three banks is handled by the program.

### File Compatibility

All files are compatible with those produced by the original version and with text files produced by other word processors compatible with DOS 2.05 or DOS 2.1.

### Printer Driver Module

Plus provides a printer driver module which allows you to construct a custom driver for your specific printer. You're required to input all those pesky control codes needed to toggle your printer's carriage return, backspace, reverse linefeed, etc. But it is easy to use and you can identify as many as nine different type styles using this feature, all of which can be accessed using the **On** command. For example, you can set up **G2** to be emphasized doubleline pitch, **G4** to be condensed superscript, **G5** to be NLQ style, and so forth.

### Mail Merge Module

This module allows you to set up a simple database which can be accessed from the main program. It can be useful in printing form letters, address labels, etc. You can use the default format or construct a custom version, but both are limited to 255 fifteen-field records. Although records can be deleted or edited, they cannot be sorted or accessed on a random basis. You can only page forward or page back.

By inserting an **Options/M** (replaces the old **Options/L** set) key sequence anywhere in your text, followed by the number of the database field desired, data can be inserted into the database and automatically inserted into your text material during the printing process.

### Proofreader Module

The Proofreader module is loaded from the main menu and utilizes a separate dictionary disk containing some 35,000 words. It can be used to check the spelling of most common words in your text material, providing you with the option of making corrections as needed, or sending a list of possible errors to your printer. You can also access the dictionary when you're not certain how to spell a given word.

although some data switching is involved. Making an enquiry on the string **MI** (for example) gave me a list of 55 words beginning with those three letters. How quickly does Proofreader function and what types of words does it fail to recognize? I've just asked for a list of possible errors in the preceding 1280 words in this article. It took 6 minutes to complete the process and gave me a list of 63 possible errors, many of which were repetitive occurrences. Among the words it did not like were Asteraster, Eleaster, keystroke, pace and super, **ngly**, **lost**.

The Proofreader module also allows you to construct a personal file of frequently used words not found in the standard dictionary, such as those mentioned above, which can be added to the current database. Now, if they would only add a thesaurus!

### Conclusion

It's time to sum up. There is no question that *SideWinder Plus* is a major improvement over the original version in new W-D editing, formatting and printing features, combined with its built-in spelling checker, mail merge and printer driver utilities make it an attractive package.

# SIDEWINDER

(c) 1984 FUTUREWARE

You are the only survivor of a team of agents sent underground to capture the latest development in helicopter technology - can you get **SIDEWINDER** out?

Also included is a **SCROLLING SCREEN EDITOR**, which enables you to construct your own **SIDEWINDER** screen for use in your game.

48K Disk £14.95/48K Cassette £9.95  
Works on all Atari Computers  
See review in this issue

Available direct or from all good  
Atari software stockists.

## FUTUREWARE

132 Gray's Inn Road, London, WC1X 8UB.  
Trade enquiries welcome.

Design the best three  
screen displays and win an  
Atari Computer & Disk Drive  
(32K+ disk Atari 1980)



# STARTING FROM BASICS

## by Captain Hacker Part Four

Welcome to the fourth part of my series of articles aimed at the beginner. Here, we try to explain how BASIC works in a way that your manuals fail to do — precisely and clearly, with frequent examples. In this issue I will continue with the theme of program-flow control.

### FOR-NEXT

In the last episode we examined how to make a program perform a given task a particular number of times. We did this using the IF THEN command by incrementing and then testing a variable to see if it had reached a particular value. If the value had not been reached we would loop back using the GOTO command. Here is a typical example of what I mean:

```
10 X=1
20 PRINT X
30 IF X=10 THEN GOTO 60
40 X=X+1
50 GOTO 20
60 STOP
```

This will print the numbers 1 to 10 on the screen. This will, of course work, but it is rather clumsy. Surely there must be a neater way of writing a loop? Well there is: the FOR-NEXT loop. Type NEW and enter the following —

```
10 FOR X=1 TO 10
20 PRINT X
30 NEXT X
60 STOP
```

Type RUN and you will see that we get printed the same result as with the previous program — and with a lot less fuss/bother!

So how does it work? Well, when BASIC encounters a FOR command, it must find at least three things:

- a) A numeric variable
- b) its starting value
- c) its finishing value

The first operation that BASIC then performs is to load the starting value into the variable, which in our example means that the value 1 is placed into the variable X. BASIC then takes a look at the FINISHING value and remembers it, and in our example this value is 10. Having done this, BASIC continues onto the next command.

The following command is of course to PRINT the contents of variable X (line 20). Since the last operation performed on the variable was to load the starting value of 1 on line 10, 1 is printed on the screen.

Now we get to the interesting bit — line 30 and the NEXT command! At this point BASIC does several things: first it must add one to the value held in the variable X, giving it a new value of 2. Then it must check to see if the value held in X is GREATER than the finishing value given in the FOR command on line 10. Since 2 is not greater than 10 instead of continuing with the subsequent line (line 40) BASIC goes back to line 10 — or to be more precise, to the FOR command. At this stage, the FOR command is not actually re-executed, but the command immediately following it is found and executed — i.e. the FOR command is now only used as a lead-in marker for the start of the loop.

The command on line 20 will again be executed, and the contents of the variable X will be printed on the screen, but this time the number 2 will appear.

So once again BASIC reaches the NEXT X command. Again, the variable X is incremented, bringing its value to 3, and it is asked to see if it is greater than the finishing value of 10. Since X is still not greater than 10, BASIC once again goes back to the command immediately after the FOR command.

This loop goes on for quite a while so rather than bore the pants off you, I will leave the routine for a while, and rejoin the BASIC as it approaches the NEXT command (on line 30) with X holding a value of 10. Well, BASIC will first increment X, making its value 11. It then checks to see if X is greater than our finishing value (10) and ... Yes, it is! (at last!)

Now BASIC terminates the FOR-NEXT LOOP, as it is called, and it

continues with the program. In our example this means that it finally reaches line 40 and executes the STOP command.

There — quite painless, wasn't it? If you are still not clear how the FOR-NEXT command operates then you should read through the previous paragraphs over and over again (and you are — because if you are still puzzled now, the following paragraph will leave you hopelessly confused! Try some FOR-NEXT loops of your own, and try different starting and finishing values.

### Variations on the Theme

Suppose that we want to create a program loop using the FOR and NEXT commands to print the numbers from 0 to 10 on the screen, but this time increasing by 2 for each loop, i.e. printing only the even numbers (0 2 4 6 8 10). Well, we can do this by using the STEP function. With the STEP function, we can decide exactly how much we want our chosen variable to be incremented by each time BASIC meets the NEXT command.

```
This is how we use STEP
10 FOR X=0 TO 10 STEP 2
20 PRINT X
30 NEXT X
60 STOP
```

Notice that the STEP is placed directly after the finishing value of the command, there is no colon or comma required, just a space. Type in the above program and RUN it: you will see the sequence 0 2 4 6 8 10 printed onto the screen.

Whatever you place after the STEP is added to the variable X each time the NEXT X command is reached, in fact this does not even have to be a whole number! You could have STEP 0.5 or STEP 0.275, or even STEP 3.9.







Continued from page 14

## Backwards

Suppose now, though, that rather than count upwards you want your loop to count backwards, say from 10 to 0. We can do this quite simply by once again using the STEP function (or modifier), but this time giving it a **NEGATIVE** value. Try this example and you will see what I mean:

```
10 FOR X=10 TO 0 STEP -1
20 PRINT X
30 NEXT X
40 STOP
```

The loop works as before except that now, when BASIC reaches the **NEXT X** command, it **SUBTRACTS** the STEP value from X, and then looks to see if the value of X is **LESS THAN** the ending value, returning to the **FOR** line if it is not.

To help you experiment with the **FOR-NEXT** loop, here is a short program with which you can mix as easily by using different starting values, finishing values, and step values. Try a starting value of 1, a finishing value of 2 and a step of 0.5 — this will give you a never ending loop!

```
20 PRINT "STARTING VALUE"
30 INPUT A
40 PRINT "FINISHING VALUE"
50 INPUT B
60 PRINT "STEP VALUE"
70 INPUT C
80 FOR X=A TO B STEP C
90 PRINT "VALUE IN VARIABLE X" = X
100 NEXT X
110 PRINT "LOOP HAS FINISHED"
110 PRINT "GOTO 10"
```

## ON X GOTO

For some reason, the **ON-GOTO** always seems to present a problem to the novice, when in fact it is one of the simplest commands available!

Perhaps the easiest way for me to explain how this command works is to show an example of what you would need to do if this command were not available. Suppose that you want to make your program jump to one of four different routines, or sections, of your program, depending upon the value of the number held in a variable. The number could have been the result of a selection from a menu of functions, and the numbers would have to be 1, 2, 3, or 4. This is how we might write the routine with which the program will decide where to go:

```
10 PRINT "CHOOSE 1,2,3 OR 4"
20 INPUT X
30 IF X=1 THEN GOTO 90
40 IF X=2 THEN GOTO 130
50 IF X=3 THEN GOTO 190
60 IF X=4 THEN GOTO 260
90 PRINT "ONE CHOSEN" GOTO 10
130 PRINT "TWO CHOSEN" GOTO 10
190 PRINT "THREE CHOSEN" GOTO 10
260 PRINT "FOUR CHOSEN" GOTO 10
```

Run the program and you will find that it works perfectly. The only trouble is though, we seem to have had to write rather a large chunk of program to just decide which line number to jump to. Take a look at lines 30 to 60, where you will see a whole cluster of **IF-THEN** commands. Just imagine how many lines this would take up if the program were to give the user 20 or even 30

different functions — you would be knee-deep in **IF-THEN** statements!

Surely I hope you agree, there has to be a simpler way. Well, surprise surprise, there is of course the **ON-GOTO** command! Take a look at the same program, but this time using the **ON-GOTO** command:

```
10 PRINT "CHOOSE 1,2,3 OR 4"
20 INPUT X
30 ON X GOTO 90,130,190,260
90 PRINT "ONE CHOSEN" GOTO 10
130 PRINT "TWO CHOSEN" GOTO 10
190 PRINT "THREE CHOSEN" GOTO 10
260 PRINT "FOUR CHOSEN" GOTO 10
```

As you can see, line 30 on its own can do the work of four that were used in the first program. You are not restricted to just four line numbers, you can have as many as you require (provided of course, that they will all fit on one command line!). You can even have an expression or calculation in place of just one variable, for example:

```
70 ON X/2 GOTO 130,80,500
```

```
or
80 ON X*3+3 GOTO 1000,980,6349
```

You might do something like this if your program drives odd number values which you will need to reduce to the 1, 2, 3, 4, 5 etc. sequence needed by the **ON-GOTO** command. Usually, however, you will probably only use this command for menu selections, so you need only ensure that the user has entered a number within the range required by your program.

## ON-GOSUB

The **ON-GOSUB** command is, as you would probably expect, virtually identical to the **ON-GOTO** which we have just covered. In case some of you are not quite sure about this, I have included (below) the **ON-GOSUB** version of our demo program. I would also like to point out that with this command it is essential that you should also observe the same rules of behavior as with the **GOSUB** command (covered in the last issue), otherwise you will get hopelessly bogged down if your program develops mysterious **BUGS**!

```
10 PRINT "CHOOSE 1,2,3 OR 4"
20 INPUT X
30 ON X GOSUB 90,130,190,260
40 GOTO 10
90 PRINT "ONE CHOSEN" RETURN
130 PRINT "TWO CHOSEN" RETURN
190 PRINT "THREE CHOSEN" RETURN
260 PRINT "FOUR CHOSEN" RETURN
```

## Tailpiece

It is a good idea to really concentrate on understanding these commands, they are the bread and butter of many BASIC programs. Experiment with different values and see what happens.

That's all for now folks! See you again in a future issue of Monitor.



# ▶▶ REVIEWS REVIEWS ▶▶ REVIEWS REVIEWS ▶▶

## Koronis Rift

Available on  
486/Meg Disk £14.99

Imagine you are a techno-savvy explorer travelling from Galaxy to Galaxy in your Surface Rover, searching for valuable loot left behind on fabulous Worlds. Suddenly you reach the legendary KORONIS RIFT, the Rift with the most valuable treasures to be collected. Your quest is to collect analyse and then keep only the most useful technological treasures you can find.

On leaving your Scoutcraft, you explore the numerous Rifts — 20 in all — which are protected by vicious Guardian Saucers — destroying these Saucers is no mean feat, they seem to appear every time you find an interesting hunk of loot!

Once you have managed to shoot down a Saucer, due to excessive radiation you call on your "top tech robot" as he or she is called, who goes into action and loots the hunk. The looted modules are then sent back to the Scoutcraft where PSYTEK — your helpful science distal systems analyser — at your request, analyses each item and then gives a value in Micromedals for efficiency and power, keep only the best to up date your own defense system careful selection is vital to the success of your mission. Wasting time and being too trigger happy can prove disastrous.

Data is shown at the top of the screen and depicts direction of track,

shield strength, as well as power reserves — use of your shields is another vital factor in how successful you become. Mapping the Rifts can help you from game to game. Strategy and tactics are very important and the "guide to the game" instructions are very helpful and clearly written.

The graphics are great, I particularly liked the view window, and PSYTEK has been animated very well indeed. There is a lot more to this game than meets the eye, the "shoot'em up" amongst us get them chance, and the more strategically minded, can also

enjoy this game and put their wits against the vicious-galactic korons.

If you need RESCUE ON FRACTULAN then you will think this is great, plenty of hunk to loot and many Rifts to explore — you can cheat and use the odd Rift if you think there are no worthwhile hunk to be plundered! — read the instructions thoroughly and they will help you to enjoy what I think is an excellent piece of computer technology.



# ▶▶ REVIEWS REVIEWS ▶ REVIEWS REVIEWS ▶▶

## Fighter Pilot

Digital Integration  
Cassette £9.95, Disk £12.95 48K (Mini)

Having spent the best part of 10 minutes working out the loading code — an ingenious if not frustrating play by Digital to stop piracy — I started on my training mission.

Flight simulation type software has never been one of my favourites. That is until now — having recently been sent a copy of *Fighter Pilot* to review, I have been greatly surprised by its complexity and, in fact, have become quite a fanatic!



## Mercenary Escape From Tang

by Newagen  
Cassette £9.95, Disk £12.95  
48K-960 48K, 1600K 1280K 64K  
Review by Matthew Teismann



After waiting a very long time for *Mercenary* to arrive from Newagen Software, it is now available and was well worth the wait! Both the disk and cassette versions of this simulation contain two separate games. The first is for the old 400-960 series and incorporates a low resolution, fast paced action game, whilst the second version is for the XL/XE series of computers and has high resolution, but rather jumpy graphics.

indicator) which gives your rate of climb and descent.

On loading the program you are confronted by 5 options: loading practice, flying training, blind landing, an up air combat practice as well as air to air combat.

The cockpit view can be replaced by a map showing where you are. The object in the final option is to defend 4 airfields with the codenames of BASE, TANGO, ZULU and DELTA. Location of the enemy is achieved by using your radar and flight computer. After combat any damage to your aircraft is shown by the change of colour of the aircraft symbol on your radar.

The noise from the cockpit sounds very authentic to me and I really was very impressed with the whole package. I shall certainly look forward to the next masterpiece from Messrs Marshall and Swift — hopefully *TOMAHAWK*!

*Fighter Pilot* originally written for the Spectrum by Dave Marshall, co-owner of Digital, who was an ex-fighter pilot and flight simulator programmer. The game was also voted "simulation of the year" for 1984.

This game is based on the USAF F15 Eagle and is described as a spectacular flight simulator. The accuracy of the instrument panel on the F15 Eagle is excellent and includes a radar compass, to find the enemy which is a flashing dot moving around your own plane, as well as VSI (vertical speed

# ▶▶ REVIEWS REVIEWS ▶ REVIEWS REVIEWS ▶▶

You take the role of a 21st-century soldier of fortune who has just crash-landed on the planet Targ. This planet has two main tribes: the Pelyers and the Mechanoids, who are both at each others' throats. Your task is to simply escape from the planet! But how? There are quite a few options open to you when you begin the simulation, for example a ship is sighted on the ground in front of you (your old ship can be seen on the horizon: nose down in the mud) which just happens to be for sale and at a reasonable price too! Should you buy it? How do you buy it? There's for you to find out!

Help is at hand however in the form of Benson, a computerised policeman who doubles as a taxi-man, mechanic, sales assistant, comedian and around helper. The total environment is represented on Benson's screen display. Using good old Benson you can buy that sleek good-looking old ship and register all of Targ. Patrol the sea for a while and you may be offered a job by one of the inhabitants which could create new problems, even all out war between the Pelyers and the Mechanoids.

Mercenary boasts truly amazing graphics especially in the fast 48K version. When comparing the speed and graphics of Mercenary to a program like Flight Simulator 2, one can begin to appreciate how complex the coding in Mercenary must be. This game could literally take weeks to complete, which adds a great sense of realism to the program. As you cannot die, you could

play Mercenary for ever!

You may be saying to yourself that flying around the sky collecting a few bits and bobs sounds easy, well you would be right, it does sound easy, but it isn't. You have to get into the underground bases to find the items that will help you to escape together with coping with a virtual war situation in which flying

spacecraft will attack you.

Mercenary is a very comprehensive adventure simulation which will give long lasting enjoyment, combined with an original design which makes it a program hard to beat.



## Sidewinder

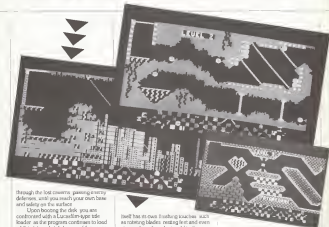
from FUTUREWARE  
£14.95 48K Disk/48K Cassette £9.95  
Advance Preview by Matthew Tydeman

FUTUREWARE are about to launch you off into a world of challenging helicopter simulation in their soon to be released **SIDEWINDER** — an action-packed program which is seen to hit your screen. This program the first arcade-type game from FUTUREWARE will give you full control over a Helicopter equipped with the latest technology. Strap yourself in and get ready for action!

You are the only survivor of a team of agents sent underground to capture the latest development in helicopter technology. The rest of your team were captured by enemy personnel in earlier missions. Your task is to fly **SIDEWINDER**



# ►►REVIEWS REVIEWS►REVIEWS REVIEWS►►



through the lost caverns, passing enemy defenses, until you reach your own base and safety on the surface.

Upon booting the disk, you are confronted with a Lucasfilm-type title loader, as the program continues to load while giving a brief glimpse of the super-techno helicopter you are about to encounter. In no time at all, the main title screen appears and is accompanied by some quite superb music, which I found mind-blowing! Begin the game and you are launched into the unknown depths of the hidden caverns.

You are coerced into a cluster of 3-Dimensional futuristic cities, landscapes and caverns through which you have to manoeuvre your super-heli by means of your joystick, by which device you also drop impact bombs and launch missiles. Moving mid-air floating mines, together with the narrow passages make the task of moving through the scrolling caverns somewhat hard to master at first, but a little experience at being behind the stick of **SIDEWINDER** soon makes the game more manageable, giving you more time to spend on navigating your ship to the end of your first mission. Your second mission then begins, with the offering of harder screens, and yet more difficult objects to successfully pass!

All screens are well detailed with waterfalls, buildings, mountains and other almost impossible objects, making each level harder and harder as you go. The **SIDEWINDER** helicopter

itself has its own floating tracks, such as rotating blades, rotating feet and even wings along the side, to add to the futuristic look of this super-sonic chopper!

There is more to the program than meets the eye. Contained on the flip side of the disk/cassette cover is the **SCROLLING EDITOR**, which was actually used to construct the game **SIDEWINDER** itself. The Editor contains features to scroll through five continuous screens (per level) which allows you to construct your own screens. In order to challenge yourself further if you find **SIDEWINDER** screens too easy! Not that you will, I think! Characters can be brought in from a pre-defined selection or by a set which you yourself make with the use of the built-in Character Editor (thought in of the press of a button). This allows you to experiment with different effects and shapes. Screen and character sets can be saved in order to incorporate them in a later program.

Colors can be changed on screen, making selection simple and convenient. A test mode can be selected so that you can try out the screen you've just made before saving it to disk, or before making a final addition to one of your recently created masterpieces. The disk version loads separate screens/characters, whilst the cassette version loads in both at the

same time from a single cassette.

The above points are perfectly good reasons for you to go out and buy this product as soon as it becomes available but, if you do require a few more deciding points, consider this: **FUTUREWARE** are offering an Atari Computer and Disk Drive to the person who designs the best 3 screen scenario with the **FUTUREWARE SIDEWINDER SCROLLING EDITOR** program! (Offer closes August 1986.)

Both **SIDEWINDER** and its Editor have proved to be simple to operate and understand. A lot of work has gone into it to make the operation simple and easy to get along with. I feel that the programmers have put a tremendous amount of work into getting the program the way it is, not just in simplicity but with the overall program.

This program, the first of many for **FUTUREWARE**, is certainly going to be a program with a difference for the Atari market. A game of this standard and variety is very rarely seen and deserves to do well when released.

Engagement, entertainment and the possibility of a free computer system cannot be had for £24.95... get it and get constructing!

# ►► REVIEWS REVIEWS ►► REVIEWS REVIEWS ►►

## Goonies

Datapak (U.S. Gold)  
Cassette £9.95 Disk £14.95 68K

It is very enjoyable to be able to review so many very good new games which have recently been released for the ATARI - about time too, did I hear - let's hope there will be many more to come.

'GOONIES' is based on the story by Steven Spielberg - say no more! If you buy the cassette version it is a good idea to have a pad and pencil handy, because the use of your tape counter (if you have one) plays a big part in the progression or otherwise to each scene or level.

The object of the game is to outwit the Funteli gang, headed by NANA and reach One-eyed Willy's Pirate Ship to find the treasure and then save the GOONIES' families. The journey through to the 8th scene is thwarted with dangers - the GOONIES have to be manoeuvred as it were, sometimes only one character will be needed to enable you to reach the next scene, but more often than not, two or three are used to negotiate the perils which can be met in the underground chambers. The graphics of the various screen displays are very clever, although if you have played 'platform' type games before, you may find them easy - the cheer and money help you through the first scene and its avoiding other objects such as bats and sneezing rocks, take you through to the steam scene - this one is not quite so easy - only manipulating the GOONIES as it were will you be able to get through each scene. No cheating it is not possible to achieve this without using the

character featured in the particular scene. Mind you, I bet there is always one who can, has or will! Turning off the steam was not at all easy.

While being a most enjoyable game to play, strategy plays a big part in how successful you can become and how quickly you reach the treasure.

A GOONIE Hand Sheet in rhyme is included to help you on your way. Each GOONIE is moved independently, so when watching characters just press your fire button. You can get three extra GOONIES after you have completed the Cannonball Chamber scene.

The graphics are excellent, but then we come to expect that with an ATARI and the music is catchy although it can

be turned off by pressing S<sup>R</sup>.

My only reservation about this game is cassette, is that each screen has to be loaded separately, which of course takes time, and if you do get 'coughed' you have to return to the previous scene and therefore have to re-load the tape to the exact spot - hence the use of your tape counter and note pad. Naturally, if you are lucky enough to have a disk drive, then loading or going back is done much quicker and easier. However, I did find the waiting a bit frustrating other than that I think the game is great and am becoming quite an addict, all got to get the treasure though!



## Alternate Reality The City

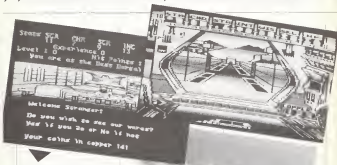
by Datapak (U.S. Gold)  
68K Disk, Price £19.95

An exciting new fantasy role playing game is to be released by U.S. Gold on the 5th of March. It is the first in the 'Alternate Reality' series and is titled 'The City'. There is to be no others in the series (which can only be played if you have a copy of City), the second is 'The Dragon' followed by 'The Arena', 'The Palace', 'The Wilderness' 'Revelation' and 'Destiny'.

The package consists of two disks, one is the main master game disk and the other is the City game disk (you obviously need the main master disk for playing the games to come). In addition you need to use a blank formatted disk of your own to create Characters. You



# ▶▶ REVIEWS REVIEWS ▶▶ REVIEWS REVIEWS ▶▶



can play without this option but you will not be able to save a game.

You will need a joystick too, as the game uses a combination of keystrokes and joystick movements. A map of the City is also included showing the Floating Gate which is in the main square surrounded by many shops, taverns, inns, banks, stables, houses and guilds. There are also secret passages which you will stumble across quite by accident.

And of course in any city you will find all sorts of inhabitants, and this one is no exception! There are the Poor and the Commoners who in the main are easy to get along with, there are Merchants, Couriers, Thugs, Noblemen, Inps and Devils, Guards, Gladiators and Assassins, Mages and Wizards, and the dreaded Night Stalker: an extremely powerful beast that no one can be certain does not exist!

Well then the City, but what are you doing there? The story goes that you were kidnapped from your safe abode on Terra Firma and whisked off to an 'Alternate Reality', all this is superbly shown in a fairly lengthy intro-sequence (you be slipped) containing some of the best gameplay I've seen for a long time. You really start when you are put down into the Gateway, which has only one exit, into 'The City of Amber's Dream'. Above your head a panel of changing numbers will set your level of Stamina, Charm, Strength, Intelligence, Wizards, Skill and Wealth as you pass into the City proper. So pick your entry time carefully to try and get the best ratings for yourself. Then off you have to do as



## Electrigrade

The English Software Company  
Cassette £8.95 Disk £12.95 +H&P

We have all been waiting a long time for a really new and exciting game for the Atari computer and English Software have come up with one. Electrigrade although by no means an easy game to play is similar to Pole Position or Baja Buggies. It is a very fast moving game giving a choice of three Continents, the tracks are similar but as the game is so fast it does not seem to matter.

Having chosen your Continent there is also a choice of "steering control envelope" - the selection depends on the type of joystick you use, the more sophisticated ones are not, in my opinion, very good for this game, a very light control is needed to speed around the track, through the tunnels and dodge the various obstacles on your way.

Each new section is reached through a tunnel but your progress is greatly hindered by bouncing missiles as well as a rocket, which drops very large columns on to the track - these

obstacles should be avoided at all costs as time is very important - each new section must be reached before the allotted time.

A display at the bottom of the screen shows a velocity indicator up to 60 wpm. When the countdown timer reaches 0 you have to start again. There are definitely 3 sections but so far I have not been able to see how many more there are - the road doesn't seem to end.

The scrolling graphics are very good - and the music is equally enjoyable.

English Software describe the game as a "Fantasy Racing Epic" - well it comes very close!

## CRACKING THE CODE

by Keith Mayhew Part Seven

### The need for Pointers

To retrace from the last issue, the two end-on-end-triplet modes we studied accessed their data through two-byte pointers stored in page zero of memory. The pre-end-on-end used the X register by adding it to the given address to create the actual address of the pointer which was then used indirectly (see Figure 1). Post-end-on-end used the Y register and subtracted through the pointer list to give the base address of the area and then added on the X register to give the final address (see Figure 3). The key

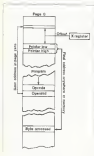


FIGURE 1 | *Flow chart of the study.*



© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 103–110

element to these addressing modes is of course their use of a pointer in accessing a byte of data. It is important that you understand their operation clearly as they are used very often when programming.

The first reason why a pointer can be used is to overcome the limitations of the 6502's absolute-indexed mode. As the index registers are 8 bits long, only 256 bytes can be accessed from any given absolute base address; the solution is to save the base address in a page zero pointer and then use an indirect-indirect mode. The idea here

To change the pointer's value to access any byte we choose. We could use either the pre-indexed or the post-indexed addressing mode, the following shows the use of the pre-indexed mode to access a table of values:

	LDX #0	Keep next index.
LOOP	LDA #POINT1X	Load a byte
	STA #POINT1X	Save it
	INC POINT1	Increment pointer
	BEQ SKIP1	Skip if not zero
	INC POINT1+1	Increment high byte
SKIP1	INC POINT2	Same for second pointer
	BEQ SKIP2	
SKIP2	INC POINT2+1	
	/ /	
	/ /	Pad with some text
	/ /	
	Branch to 'LOOP	Go back if more to count

POINT1 and POINT2 are assumed to be in page zero and to have been initialized to point to the start of two areas of memory, or tables. The X register is kept at zero, otherwise we wouldn't be accessing the same pointer each time! Each time around the loop the 2 two byte pointers are incremented by one and then some test is performed followed by a branch back to the start if more is to be processed. If not it is necessary

*This time we conclude our study of the indexed-indirect modes and find out why they are useful! Having done that we will have covered all the 6502's instructions and addressing modes and will be free to explore the many features of the ATARI computers, further articles will be concerned with using the machine's resources from assembly language, such as controlling the graphics and sound hardware.*

The operating system will also be covered to see how we can perform input/output operations; we will also be studying the use of interrupts and how the operating system employs them for a lot of its work.

move any size blocks of memory. The program could have used the Y register in a similar way, but the post-indexed mode can be faster and more flexible.

```
LDY #0 ;Start with zero
LOOP LDA (POINT1),Y Load byte
STA (POINT2),Y Save byte
INY ;Increment Y
BNE SKIP1 ;Skip if not zero
INC POINT1+1 ;Else inc. high bytes of pointers
INC POINT2+1
SKIP1 / / Perform second test.
/ /
/ /
"branch to" LOOP ;Go back if more to copy
```

The program will work a lot faster because lots of memory accesses are being performed in the loop. This time we increment the Y register instead of the low byte of the pointer to copy up to 256 bytes, once the Y register goes back to zero we simply increment the two high bytes of the pointers to move along another 256 bytes into each table. As you might have guessed, this post-indexed mode is used for more often than its pre-indexed counterpart on the basis of saving space and bytes in the program; it is useful consequence is that the Y register may be re-loaded with a new index, used, and then the old index restored, without having to touch the pointer.

## Keeping Track

The second reason for using pointers, even when accessing 256 bytes or less, is that tables of data may not be in a fixed position in memory. Consider the problem of building several tables in memory, each of an undetermined size. You could start each table at a fixed address and then hope that one table doesn't overwrite the next... even if it does! you could well be wasting a lot of memory unnecessarily. The solution would be to set the pointer to the first table at the start of available memory, once built, the next free address would be stored as the pointer to the second table and so on, in this way the memory will be used more efficiently. A similar problem is when a subroutine has to be able to work on an arbitrary table; the calling program could simply save its pointer to its table into the subroutine pointer and then call the subroutine.

Pointers are used by the Atari operating system for one main reason: the amount of memory in the system could be different. For instance, the screen memory is a 16K machine starts at a different place than that in a 48K machine, as a consequence a pointer is

kept to the start of the screen; it then doesn't matter to the operating system where the screen is actually located, it will always access it correctly. It should be stressed that it is bad programming practice to assume that some area of memory is fixed when it could in fact be in a different place (I have seen programs that were written for a 48K machine and when run on a 16K machine, although there was enough memory, a blank screen appeared because the screen memory had moved!).

## Accessing the Screen

The reason why I have chosen to start with the screen is that it is an effective way of demonstrating the indexed-indirect accessing with indirect vector results for you to relate to, and is a practical situation.

The shortest program I left you with last time (Listing 4) when run showed all the possible 256 characters of the top of a graphics 0 screen. It did this by using the operating system pointer to the start of the screen, which is located at \$8 hex, the Y register was used to index up to 256 locations into the screen, and at each location it stored the index value by transferring the Y into the accumulator.

## The Screen Layout

Each of the standard graphic and text modes display a fixed number of bytes per line and fixed number of lines down the screen. In common with all these modes is a simple mapping from a contiguous memory area onto the physical screen. The first byte of the memory area is mapped to the top left hand part of the screen; the next byte is mapped to the right of the last one, and so on until one complete line is displayed; the consecutive bytes are then mapped to the start of a left side of the next line, and so on, until the whole picture is displayed. Figure 3 shows the mapping for a 40 byte by 24 line display, such as graphics 0, with the byte offsets to each screen position, from the start of the screen memory, in bytes, shown inside the positions.

Screen address memory	Byte offset into screen memory to position shown
Line 1	0 1 2 ... 39 40
Line 2	40 41 42 ... 79 80
Line 3	80 81 82 ... 119 120
Line 22	840 841 842 ... 879 880
Line 23	880 881 882 ... 919 920
Line 24	920 921 922 ... 959 960

Figure 3 Screen Layout

## Character & Bit Mapping

The Atari hardware allows for two methods of displaying screens: character and bit mapped; the difference being how the bytes from memory are displayed: both have the effect of turning a set of picture elements or physical pixels, into a particular colour on the screen. A pixel is the smallest addressable unit of the display.

In a character mapped mode, such as graphics 0, each byte read from memory causes a matrix of physical pixels to be set, in this case it covers an 8 by 8 pixels area. The exact pattern of the character is determined by the character set being used.

A bit mapped mode splits the byte into a number of bit groups, or fields, each with the same number of bits, e.g. eight by 1 bit, four by 2 bits, two by 4 bits or one whole byte. Each group of bits then determines the state of one logical pixel on the screen. Note that a logical pixel is a group of physical pixels, which can be treated as one entity because only the hardware addresses the physical pixels we still subdivide the logical pixels into address to just pixels or points. The distinction is not very important just as long as you understand that a group of bits in a byte are controlling the state of one area, of some dimension, on the screen. For example graphics 8 has 40 bytes per line and 192 lines per screen; each byte is split into eight bits, each bit either turns a pixel on or off (one colour or another) giving 40\*8, or 320 pixels per line by 192 down.

## Using a Bit Mapped Screen

Listing 1 accesses the bit mapped screen of graphics 8 via the operating system pointer at \$8 hex, using for all modes, to produce a random moving pattern.

CONSOLE and RANDOM are labels set up to access special locations in the hardware; all you need know for now is that reading from CONSOLE gives the status of the START, SELECT and OPTION keys and reading from RANDOM produces a 'random' 8 bit number. \$A7FASC is the name given to the pointer to the start of the screen memory. PZERO and TEMP are variables used by the program, they have been assigned the values of \$CB and \$CD respectively. Straight forward equates could have been given e.g. PZERO = \$CB, but instead I have set the location counter to \$CB and defined the label PZERO. " = " + 2 assigns the value of the location counter " + 2 plus two back to the location counter, thus moving it forward by two, the same occurs for TEMP giving it the value of \$CD. The forward movement of the location counter is often referred to as 'define ahead' and on some assemblers there is a special directive to perform this

```

0100 CONTROL = 40817 (Load START/SELECT/OPTION.
0110 RANDOM = 40204 (Random number generator.
0120 SAVWSC = 408 (Screen pointer.
0130 = 408 (Start of variables.
0140 ACOORD = 402 (Pointer to screen.
0150 TEMP = 402 (Temporary locations.
0160 = 40400 (Start of program.
0170 ALA (Clean up stack.
0180 START LDA SAVWSC (Load 1st byte
0190 STA PZERO % of pointer.
0200 LDA SAVWSC+1 (Store with high.
0210 STA ACOORD+1
0220 RNDM LDA RANDOM (Get random number.
0230 CMP RNDM (If 192 or more
0240 CCS RNDM % get another one.
0250 STA TEMP (Store as low byte of Y.
0260 LDA 40 (
0270 STA TEMP+1 (Store high byte.
0280 LDA 82 (Count of 3.
0290 MUL10 ROL TEMP (Multiply) by 3...
0300 ROL TEMP+1
0310 ROL (Dec. count and branch
0320 BNC MUL10 % to give 8 times TEMP.
0330 LDA ACOORD (Add low byte to ACOORD.
0340 CLC
0350 ADC TEMP

```

```

0360 STA PZERO (Add high byte.
0370 LDA RNDM+1 (Multiply TEMP
0380 ADC RNDM+1 % by 4
0390 STA ACOORD+1 (to give 32 times
0400 ROL TEMP+1 (as total.
0410 LDA ACOORD (Add to PZERO again.
0420 CLC
0430 ADC TEMP
0440 STA ACOORD
0450 LDA ACOORD+1 (Add high byte
0460 ADC TEMP+1 (to give 40th value.
0470 STA ACOORD+1
0480 RNDM (Another random number,
0490 (Store top 2 bits.
0500 AND 4096 (If 40 or more
0510 CCS RNDM % get another.
0520 TRV (Same as index into line.
0530 LDA RANDOM (Another random number.
0540 STA PZERO (Store on screen.
0550 LDA CCOORD (First letters.
0560 CMP 47 (All up?
0570 BNC START (Yes - go again.
0580 RTS (Line returns to BASIC.

```

#### Listing 1

operation. The advantages of this method are that you can instantly see how many bytes are intended for each storage area, two in this case, and that you do not have to do any calculations for the value of each label. Lastly the whole block can be moved to another area by simply changing the origin value before the labels — a great time saver if there are a lot of variables!

Now onto the program. The code starts at 400 here, the first thing done is to remove the byte BASIC gave us on the stack. So we need to change the contents of SAVWSC, the two-byte pointer is copied into PZERO and PZERO+1. A random number is then loaded into the accumulator and tested against the number of lines down our screen — if it is greater than or equal to 192 then another number is picked, when the loop is finished A will contain a number between 0 and 191 inclusive. This Y co-ordinate is then saved on TEMP and the high byte of TEMP (TEMP+1) is moved, the code from line 280 to 400 then multiplies this Y value by 40, the number of bytes per line, to give the offset to the start of the given line, which is then added to our pointer to the start of the screen, PZERO.

For speed and convenience the method by which TEMP is multiplied by 40 is to multiply it by 8, add it onto PZERO, and then to multiply TEMP by a further 4 to give a total of 32 times which

is also added to PZERO. The 8 times TEMP plus the 32 times TEMP gives the 40 times TEMP we needed to add PZERO. The multiplications are all

```

02 10 015 RES+110
03 20 LDA+10000:TEMP (Y)=40:TEMP+120
04 30 READ RNDM,CORCOORD+0
05 40 FOR I=1 TO 35 STOP 3
06 50 IF RNDM<191:110-40:TEMP+ADC:RNDM+1:1,1:10-40
07 60 RNDM=COR-T+100:16:16+32-T+COR:16:10
08 70 SUM=SUM+RNDM:ACORD=START+2,000:J=J+1:ACORD+1
09 80 IF SUM=COORDIN THEN LINE,LINE+1:STOP 3
10 90 "Check error on this line!"
11 05 LOST LINE:END
12 10 PRINT "Data is wrong."
13 10000 DATA J=0:SUM=0:COORDIN=0
14 10010 DATA COR=0:COORDIN=0:1,1,15
15 10020 DATA COR=0:COORDIN=0:1,1,15
16 10030 DATA COR=0:COORDIN=0:1,1,15
17 10040 DATA COR=0:COORDIN=0:1,1,15
18 10050 DATA COR=0:COORDIN=0:1,1,15
19 10060 DATA COR=0:COORDIN=0:1,1,15
20 10070 DATA COR=0:COORDIN=0:1,1,15
21 10080 DATA COR=0:COORDIN=0:1,1,15
22 10090 DATA COR=0:COORDIN=0:1,1,15
23 10100 DATA COR=0:COORDIN=0:1,1,15

```

#### Listing 2

performed on TEMP and TEMP+1 using shift and rotate operators multiplying by two each time.

Another random number is now loaded, but this time it has to be between 0 and 39, so we compare it to 40 and get another number if the carry is set. To speed the selection process, the top two bits of the random number are moved by AND with a mask of 00111111 or 39F, giving a number between 0 and 63, thus some of a chance it will be in the right range (15% versus 63% in theory). This value is used as the byte index into the selected line and is moved to the Y register. Yet another random number is loaded and the 8 bit pattern is stored into the selected byte of the screen, pointed at by PZERO and offset by the Y register. By changing the byte, 8 pixels will set to on or off in that area. Lastly the program loads the value of CONTROL and checks it against 7, if it is equal the program starts again, otherwise it stops and returns to BASIC.

To run the program either type in the BASIC program, Listing 2, or assemble the source code of Listing 1 and load the object code. Once the program is loaded, from 1034 onwards, it can be run by typing: COR 24 X=USR(1234) from BASIC. If the screen fills up with random rubbish don't worry, that means the program is working! To check a program when the START, SELECT or OPTION key

## BASIC Loaders

To finish this time we have three BASIC programs, the one last time read a binary file from disk, or tape into memory; these two are short utilities to create data and to read data for use in a BASIC program.

Listing 3 asks for a device to write in file to, again, for convenience C:, and for disk D: followed by a filename. It will then ask you for four more decimal numbers. Assuming you have got assembled code sitting in memory, the program will create a file containing line numbers and DATA statements. The data in the area you specified by the start and end addresses will be placed after the DATA statements and the line numbers will start at the number you gave and increment by the number you gave. The program will then tell you how many bytes it has read from memory. The file, once created, can then be ENTERED over any existing BASIC program which can then read in the data, such as Listing 6, note that if you know how many bytes are to be read it is easier to use a FOR/NEXT loop.

Listings 5 and 6 are similar to those in 3 and 4. Listing 5 makes a file of DATA statements but puts its data in strings of hex characters, which can then be read by a program such as that in Listing 6. If you haven't got an assembler then listing 6 is particularly useful as you will be able to write hex characters on the DATA statement lines in the following way:

```
10000 DATA #00FF#6829#06#4BC579#08
```

The twenty-four characters represent the twelve hex numbers to be placed in consecutive locations; you could write more or less characters per line, but this is the number of data which Listing 6 processes. Lastly, Listing 7 creates a file containing a string dimension and then some string assignments. If any quotes or carriage return characters are found, then separate assignments are made for them using the CHR\$ function. If you use the method you must make sure that the program, which is placed in the string, MCR\$ is relocatable; e.g. there are no jump or pump to sub-routine instructions as the program will crash. To call the code in a string use the string address function

```
X%=USR1ADDR(MCR$)
```

Hopefully I have covered as much short utilities in BASIC for most of your purposes and it shouldn't be too difficult to modify the routines to suit your own needs.

Next time we will start a detailed examination of the hardware and operating system facilities.

```
10 10 DIM F$LEN(14)
20 20 ? "Please enter device/file name"
30 30 ? "for output..."
40 40 INPUT F$
50 50 OPEN #1,J,I,F$
60 60 ? "Please enter start address..."
70 70 INPUT ST
80 80 ? "Please enter end address..."
90 90 INPUT EN
100 100 ? "Please enter starting line..."
110 110 INPUT L1N
120 120 ? "Please enter line increment..."
130 130 INPUT INC
140 140 CLOSE #1
150 150 FOR I=ST TO EN
160 160 IF C=0 THEN PRINT #1,L1N;" DATA "
170 170 IF C=0 OR I=EN THEN PRINT #1,PEEK
180 180 C1+ST*2
190 190 PRINT #1,PEEK(C1)," "
200 200 C=C+1:IF C=0 THEN C=255:IN=
210 210 INC
220 220 NEXT I
230 230 CLOSE #1
240 240 ? "Number of data = "C$=ST*2
```

Listing 3

```
50 50 TRAP #0:GOTO 1000
60 60 READ NUM:FOR L1N=NUM
70 70 L1N=L1N+1:GOTO 20
80 80 END
```

Listing 4

```
10 10 DIM F$LEN(14),MCR$(255),C$MCR$(M)
20 20 C$MCR$="002456789ABCDEF"
30 30 ? "Please enter device/file name"
40 40 ? "for output..."
50 50 INPUT F$
60 60 OPEN #1,J,I,F$
70 70 ? "Please enter start address..."
80 80 INPUT ST
90 90 ? "Please enter end address..."
100 100 INPUT EN
110 110 ? "Please enter starting line..."
120 120 INPUT L1N
130 130 ? "Please enter line increment..."
140 140 INPUT INC
150 150 L1N=L1N+1:GOTO 60
160 160 MCR$=CHR$(M)
170 170 IF MCR$(M) THEN MCR$(M)
180 180 FOR I=1 TO 255:STEP 2
190 190 L1=L1+PEEK(L1N)
200 200 C1=INT(L1/16)+256-I:R1=
210 210 MCR$(I),C1+CONV(R1+1,R1+1+PEEK(I)
220 220 R1+1)+CONV(R1+1,R1+1)
230 230 L1N=L1N+1
240 240 NEXT I
250 250 PRINT #1,L1N;" DATA "MCR$
260 260 L1N=L1N+INC:L1N=L1N-12
270 270 IF L1N=EN THEN 180
280 280 ? "Number of data = "C$=ST*2
290 290 CLOSE #1
```

Listing 5

```
10 10 DIM MCR$(255)
20 20 TRAP #0:GOTO 1000
30 30 READ MCR$
40 40 FOR I=1 TO LEN(MCR$):STEP 2
50 50 C1=CONV(PEEK(I),16)+48:R1=CONV(PEEK(I)
60 60 I+1,16)+48
70 70 MCR$=MCR$(I)+CHR$(C1)+CHR$(R1)
80 80 NEXT I
90 90 PRINT #1,C$=MCR$(C$+1)
100 100 NEXT I
110 110 GOTO 20
120 120 END
```

Listing 6

```
10 10 DIM F$LEN(14),C$MCR$(255),C$MCR$(M)
20 20 ? "Please enter device/file name"
30 30 ? "for output..."
40 40 INPUT F$
50 50 OPEN #1,J,I,F$
60 60 ? "Please enter start address..."
70 70 INPUT ST
80 80 ? "Please enter end address..."
90 90 INPUT EN
100 100 ? "Please enter starting line..."
110 110 INPUT L1N
120 120 ? "Please enter line increment..."
130 130 INPUT INC
140 140 L1N=L1N+1
150 150 PRINT #1,L1N;" DATA "MCR$(L1N)
160 160 L1N=L1N+PEEK(L1N)+1
170 170 C1=INT(L1N/16)+256-L1N+1:R1=
180 180 IF MCR$(L1N) THEN MCR$(L1N)
190 190 PRINT #1,L1N;" MCR$("PEEK(C1)," "PEEK(
200 200 C1+1)
210 210 FOR I=PEEK TO PEEK
220 220 C1=CONV(PEEK(I),16)
230 230 IF I=PEEK THEN ST=ST+1:GOTO 160
240 240 IF I=PEEK THEN C1=CONV(PEEK(I),16)+
250 250 48
260 260 FOR I=ST TO EN
270 270 P1=PEEK(I):P2=PEEK(I+1)
280 280 P3=PEEK(I+2):P4=PEEK(I+3)
290 290 P5=PEEK(I+4):P6=PEEK(I+5)
300 300 P7=PEEK(I+6):P8=PEEK(I+7)
310 310 P9=PEEK(I+8):P10=PEEK(I+9)
320 320 P11=PEEK(I+10):P12=PEEK(I+11)
330 330 P13=PEEK(I+12):P14=PEEK(I+13)
340 340 P15=PEEK(I+14):P16=PEEK(I+15)
350 350 P17=PEEK(I+16):P18=PEEK(I+17)
360 360 P19=PEEK(I+18):P20=PEEK(I+19)
370 370 P21=PEEK(I+20):P22=PEEK(I+21)
380 380 P23=PEEK(I+22):P24=PEEK(I+23)
390 390 P25=PEEK(I+24):P26=PEEK(I+25)
400 400 P27=PEEK(I+26):P28=PEEK(I+27)
410 410 P29=PEEK(I+28):P30=PEEK(I+29)
420 420 P31=PEEK(I+30):P32=PEEK(I+31)
430 430 P33=PEEK(I+32):P34=PEEK(I+33)
440 440 P35=PEEK(I+34):P36=PEEK(I+35)
450 450 P37=PEEK(I+36):P38=PEEK(I+37)
460 460 P39=PEEK(I+38):P40=PEEK(I+39)
470 470 P41=PEEK(I+40):P42=PEEK(I+41)
480 480 P43=PEEK(I+42):P44=PEEK(I+43)
490 490 P45=PEEK(I+44):P46=PEEK(I+45)
500 500 P47=PEEK(I+46):P48=PEEK(I+47)
510 510 P49=PEEK(I+48):P50=PEEK(I+49)
520 520 P51=PEEK(I+50):P52=PEEK(I+51)
530 530 P53=PEEK(I+52):P54=PEEK(I+53)
540 540 P55=PEEK(I+54):P56=PEEK(I+55)
550 550 P57=PEEK(I+56):P58=PEEK(I+57)
560 560 P59=PEEK(I+58):P60=PEEK(I+59)
570 570 P61=PEEK(I+60):P62=PEEK(I+61)
580 580 P63=PEEK(I+62):P64=PEEK(I+63)
590 590 P65=PEEK(I+64):P66=PEEK(I+65)
600 600 P67=PEEK(I+66):P68=PEEK(I+67)
610 610 P69=PEEK(I+68):P70=PEEK(I+69)
620 620 P71=PEEK(I+70):P72=PEEK(I+71)
630 630 P73=PEEK(I+72):P74=PEEK(I+73)
640 640 P75=PEEK(I+74):P76=PEEK(I+75)
650 650 P77=PEEK(I+76):P78=PEEK(I+77)
660 660 P79=PEEK(I+78):P80=PEEK(I+79)
670 670 P81=PEEK(I+80):P82=PEEK(I+81)
680 680 P83=PEEK(I+82):P84=PEEK(I+83)
690 690 P85=PEEK(I+84):P86=PEEK(I+85)
700 700 P87=PEEK(I+86):P88=PEEK(I+87)
710 710 P89=PEEK(I+88):P90=PEEK(I+89)
720 720 P91=PEEK(I+90):P92=PEEK(I+91)
730 730 P93=PEEK(I+92):P94=PEEK(I+93)
740 740 P95=PEEK(I+94):P96=PEEK(I+95)
750 750 P97=PEEK(I+96):P98=PEEK(I+97)
760 760 P99=PEEK(I+98):P100=PEEK(I+99)
770 770 P101=PEEK(I+100):P102=PEEK(I+101)
780 780 P103=PEEK(I+102):P104=PEEK(I+103)
790 790 P105=PEEK(I+104):P106=PEEK(I+105)
800 800 P107=PEEK(I+106):P108=PEEK(I+107)
810 810 P109=PEEK(I+108):P110=PEEK(I+109)
820 820 P111=PEEK(I+110):P112=PEEK(I+111)
830 830 P113=PEEK(I+112):P114=PEEK(I+113)
840 840 P115=PEEK(I+114):P116=PEEK(I+115)
850 850 P117=PEEK(I+116):P118=PEEK(I+117)
860 860 P119=PEEK(I+118):P120=PEEK(I+119)
870 870 P121=PEEK(I+120):P122=PEEK(I+121)
880 880 P123=PEEK(I+122):P124=PEEK(I+123)
890 890 P125=PEEK(I+124):P126=PEEK(I+125)
900 900 P127=PEEK(I+126):P128=PEEK(I+127)
910 910 P129=PEEK(I+128):P130=PEEK(I+129)
920 920 P131=PEEK(I+130):P132=PEEK(I+131)
930 930 P133=PEEK(I+132):P134=PEEK(I+133)
940 940 P135=PEEK(I+134):P136=PEEK(I+135)
950 950 P137=PEEK(I+136):P138=PEEK(I+137)
960 960 P139=PEEK(I+138):P140=PEEK(I+139)
970 970 P141=PEEK(I+140):P142=PEEK(I+141)
980 980 P143=PEEK(I+142):P144=PEEK(I+143)
990 990 P145=PEEK(I+144):P146=PEEK(I+145)
1000 1000 P147=PEEK(I+146):P148=PEEK(I+147)
1010 1010 P149=PEEK(I+148):P150=PEEK(I+149)
1020 1020 P151=PEEK(I+150):P152=PEEK(I+151)
1030 1030 P153=PEEK(I+152):P154=PEEK(I+153)
1040 1040 P155=PEEK(I+154):P156=PEEK(I+155)
1050 1050 P157=PEEK(I+156):P158=PEEK(I+157)
1060 1060 P159=PEEK(I+158):P160=PEEK(I+159)
1070 1070 P161=PEEK(I+160):P162=PEEK(I+161)
1080 1080 P163=PEEK(I+162):P164=PEEK(I+163)
1090 1090 P165=PEEK(I+164):P166=PEEK(I+165)
1100 1100 P167=PEEK(I+166):P168=PEEK(I+167)
1110 1110 P169=PEEK(I+168):P170=PEEK(I+169)
1120 1120 P171=PEEK(I+170):P172=PEEK(I+171)
1130 1130 P173=PEEK(I+172):P174=PEEK(I+173)
1140 1140 P175=PEEK(I+174):P176=PEEK(I+175)
1150 1150 P177=PEEK(I+176):P178=PEEK(I+177)
1160 1160 P179=PEEK(I+178):P180=PEEK(I+179)
1170 1170 P181=PEEK(I+180):P182=PEEK(I+181)
1180 1180 P183=PEEK(I+182):P184=PEEK(I+183)
1190 1190 P185=PEEK(I+184):P186=PEEK(I+185)
1200 1200 P187=PEEK(I+186):P188=PEEK(I+187)
1210 1210 P189=PEEK(I+188):P190=PEEK(I+189)
1220 1220 P191=PEEK(I+190):P192=PEEK(I+191)
1230 1230 P193=PEEK(I+192):P194=PEEK(I+193)
1240 1240 P195=PEEK(I+194):P196=PEEK(I+195)
1250 1250 P197=PEEK(I+196):P198=PEEK(I+197)
1260 1260 P199=PEEK(I+198):P200=PEEK(I+199)
1270 1270 P201=PEEK(I+200):P202=PEEK(I+201)
1280 1280 P203=PEEK(I+202):P204=PEEK(I+203)
1290 1290 P205=PEEK(I+204):P206=PEEK(I+205)
1300 1300 P207=PEEK(I+206):P208=PEEK(I+207)
1310 1310 P209=PEEK(I+208):P210=PEEK(I+209)
1320 1320 P211=PEEK(I+210):P212=PEEK(I+211)
1330 1330 P213=PEEK(I+212):P214=PEEK(I+213)
1340 1340 P215=PEEK(I+214):P216=PEEK(I+215)
1350 1350 P217=PEEK(I+216):P218=PEEK(I+217)
1360 1360 P219=PEEK(I+218):P220=PEEK(I+219)
1370 1370 P221=PEEK(I+220):P222=PEEK(I+221)
1380 1380 P223=PEEK(I+222):P224=PEEK(I+223)
1390 1390 P225=PEEK(I+224):P226=PEEK(I+225)
1400 1400 P227=PEEK(I+226):P228=PEEK(I+227)
1410 1410 P229=PEEK(I+228):P230=PEEK(I+229)
1420 1420 P231=PEEK(I+230):P232=PEEK(I+231)
1430 1430 P233=PEEK(I+232):P234=PEEK(I+233)
1440 1440 P235=PEEK(I+234):P236=PEEK(I+235)
1450 1450 P237=PEEK(I+236):P238=PEEK(I+237)
1460 1460 P239=PEEK(I+238):P240=PEEK(I+239)
1470 1470 P241=PEEK(I+240):P242=PEEK(I+241)
1480 1480 P243=PEEK(I+242):P244=PEEK(I+243)
1490 1490 P245=PEEK(I+244):P246=PEEK(I+245)
1500 1500 P247=PEEK(I+246):P248=PEEK(I+247)
1510 1510 P249=PEEK(I+248):P250=PEEK(I+249)
1520 1520 P251=PEEK(I+250):P252=PEEK(I+251)
1530 1530 P253=PEEK(I+252):P254=PEEK(I+253)
1540 1540 P255=PEEK(I+254):P256=PEEK(I+255)
1550 1550 P257=PEEK(I+256):P258=PEEK(I+257)
1560 1560 P259=PEEK(I+258):P260=PEEK(I+259)
1570 1570 P261=PEEK(I+260):P262=PEEK(I+261)
1580 1580 P263=PEEK(I+262):P264=PEEK(I+263)
1590 1590 P265=PEEK(I+264):P266=PEEK(I+265)
1600 1600 P267=PEEK(I+266):P268=PEEK(I+267)
1610 1610 P269=PEEK(I+268):P270=PEEK(I+269)
1620 1620 P271=PEEK(I+270):P272=PEEK(I+271)
1630 1630 P273=PEEK(I+272):P274=PEEK(I+273)
1640 1640 P275=PEEK(I+274):P276=PEEK(I+275)
1650 1650 P277=PEEK(I+276):P278=PEEK(I+277)
1660 1660 P279=PEEK(I+278):P280=PEEK(I+279)
1670 1670 P281=PEEK(I+280):P282=PEEK(I+281)
1680 1680 P283=PEEK(I+282):P284=PEEK(I+283)
1690 1690 P285=PEEK(I+284):P286=PEEK(I+285)
1700 1700 P287=PEEK(I+286):P288=PEEK(I+287)
1710 1710 P289=PEEK(I+288):P290=PEEK(I+289)
1720 1720 P291=PEEK(I+290):P292=PEEK(I+291)
1730 1730 P293=PEEK(I+292):P294=PEEK(I+293)
1740 1740 P295=PEEK(I+294):P296=PEEK(I+295)
1750 1750 P297=PEEK(I+296):P298=PEEK(I+297)
1760 1760 P299=PEEK(I+298):P300=PEEK(I+299)
1770 1770 P301=PEEK(I+300):P302=PEEK(I+301)
1780 1780 P303=PEEK(I+302):P304=PEEK(I+303)
1790 1790 P305=PEEK(I+304):P306=PEEK(I+305)
1800 1800 P307=PEEK(I+306):P308=PEEK(I+307)
1810 1810 P309=PEEK(I+308):P310=PEEK(I+309)
1820 1820 P311=PEEK(I+310):P312=PEEK(I+311)
1830 1830 P313=PEEK(I+312):P314=PEEK(I+313)
1840 1840 P315=PEEK(I+314):P316=PEEK(I+315)
1850 1850 P317=PEEK(I+316):P318=PEEK(I+317)
1860 1860 P319=PEEK(I+318):P320=PEEK(I+319)
1870 1870 P321=PEEK(I+320):P322=PEEK(I+321)
1880 1880 P323=PEEK(I+322):P324=PEEK(I+323)
1890 1890 P325=PEEK(I+324):P326=PEEK(I+325)
1900 1900 P327=PEEK(I+326):P328=PEEK(I+327)
1910 1910 P329=PEEK(I+328):P330=PEEK(I+329)
1920 1920 P331=PEEK(I+330):P332=PEEK(I+331)
1930 1930 P333=PEEK(I+332):P334=PEEK(I+333)
1940 1940 P335=PEEK(I+334):P336=PEEK(I+335)
1950 1950 P337=PEEK(I+336):P338=PEEK(I+337)
1960 1960 P339=PEEK(I+338):P340=PEEK(I+339)
1970 1970 P341=PEEK(I+340):P342=PEEK(I+341)
1980 1980 P343=PEEK(I+342):P344=PEEK(I+343)
1990 1990 P345=PEEK(I+344):P346=PEEK(I+345)
2000 2000 P347=PEEK(I+346):P348=PEEK(I+347)
2010 2010 P349=PEEK(I+348):P350=PEEK(I+349)
2020 2020 P351=PEEK(I+350):P352=PEEK(I+351)
2030 2030 P353=PEEK(I+352):P354=PEEK(I+353)
2040 2040 P355=PEEK(I+354):P356=PEEK(I+355)
2050 2050 P357=PEEK(I+356):P358=PEEK(I+357)
2060 2060 P359=PEEK(I+358):P360=PEEK(I+359)
2070 2070 P361=PEEK(I+360):P362=PEEK(I+361)
2080 2080 P363=PEEK(I+362):P364=PEEK(I+363)
2090 2090 P365=PEEK(I+364):P366=PEEK(I+365)
2100 2100 P367=PEEK(I+366):P368=PEEK(I+367)
2110 2110 P369=PEEK(I+368):P370=PEEK(I+369)
2120 2120 P371=PEEK(I+370):P372=PEEK(I+371)
2130 2130 P373=PEEK(I+372):P374=PEEK(I+373)
2140 2140 P375=PEEK(I+374):P376=PEEK(I+375)
2150 2150 P377=PEEK(I+376):P378=PEEK(I+377)
2160 2160 P379=PEEK(I+378):P380=PEEK(I+379)
2170 2170 P381=PEEK(I+380):P382=PEEK(I+381)
2180 2180 P383=PEEK(I+382):P384=PEEK(I+383)
2190 2190 P385=PEEK(I+384):P386=PEEK(I+385)
2200 2200 P387=PEEK(I+386):P388=PEEK(I+387)
2210 2210 P389=PEEK(I+388):P390=PEEK(I+389)
2220 2220 P391=PEEK(I+390):P392=PEEK(I+391)
2230 2230 P393=PEEK(I+392):P394=PEEK(I+393)
2240 2240 P395=PEEK(I+394):P396=PEEK(I+395)
2250 2250 P397=PEEK(I+396):P398=PEEK(I+397)
2260 2260 P399=PEEK(I+398):P400=PEEK(I+399)
2270 2270 P401=PEEK(I+400):P402=PEEK(I+401)
2280 2280 P403=PEEK(I+402):P404=PEEK(I+403)
2290 2290 P405=PEEK(I+404):P406=PEEK(I+405)
2300 2300 P407=PEEK(I+406):P408=PEEK(I+407)
2310 2310 P409=PEEK(I+408):P410=PEEK(I+409)
2320 2320 P411=PEEK(I+410):P412=PEEK(I+411)
2330 2330 P413=PEEK(I+412):P414=PEEK(I+413)
2340 2340 P415=PEEK(I+414):P416=PEEK(I+415)
2350 2350 P417=PEEK(I+416):P418=PEEK(I+417)
2360 2360 P419=PEEK(I+418):P420=PEEK(I+419)
2370 2370 P421=PEEK(I+420):P422=PEEK(I+421)
2380 2380 P423=PEEK(I+422):P424=PEEK(I+423)
2390 2390 P425=PEEK(I+424):P426=PEEK(I+425)
2400 2400 P427=PEEK(I+426):P428=PEEK(I+427)
2410 2410 P429=PEEK(I+428):P430=PEEK(I+429)
2420 2420 P431=PEEK(I+430):P432=PEEK(I+431)
2430 2430 P433=PEEK(I+432):P434=PEEK(I+433)
2440 2440 P435=PEEK(I+434):P436=PEEK(I+435)
2450 2450 P437=PEEK(I+436):P438=PEEK(I+437)
2460 2460 P439=PEEK(I+438):P440=PEEK(I+439)
2470 2470 P441=PEEK(I+440):P442=PEEK(I+441)
2480 2480 P443=PEEK(I+442):P444=PEEK(I+443)
2490 2490 P445=PEEK(I+444):P446=PEEK(I+445)
2500 2500 P447=PEEK(I+446):P448=PEEK(I+447)
2510 2510 P449=PEEK(I+448):P450=PEEK(I+449)
2520 2520 P451=PEEK(I+450):P452=PEEK(I+451)
2530 2530 P453=PEEK(I+452):P454=PEEK(I+453)
2540 2540 P455=PEEK(I+454):P456=PEEK(I+455)
2550 2550 P457=PEEK(I+456):P458=PEEK(I+457)
2560 2560 P459=PEEK(I+458):P460=PEEK(I+459)
2570 2570 P461=PEEK(I+460):P462=PEEK(I+461)
2580 2580 P463=PEEK(I+462):P464=PEEK(I+463)
2590 2590 P465=PEEK(I+464):P466=PEEK(I+465)
2600 2600 P467=PEEK(I+466):P468=PEEK(I+467)
2610 2610 P469=PEEK(I+468):P470=PEEK(I+469)
2620 2620 P471=PEEK(I+470):P472=PEEK(I+471)
2630 2630 P473=PEEK(I+472):P474=PEEK(I+473)
2640 2640 P475=PEEK(I+474):P476=PEEK(I+475)
2650 2650 P477=PEEK(I+476):P478=PEEK(I+477)
2660 2660 P479=PEEK(I+478):P480=PEEK(I+479)
2670 2670 P481=PEEK(I+480):P482=PEEK(I+481)
2680 2680 P483=PEEK(I+482):P484=PEEK(I+483)
2690 2690 P485=PEEK(I+484):P486=PEEK(I+485)
2700 2700 P487=PEEK(I+486):P488=PEEK(I+487)
2710 2710 P489=PEEK(I+488):P490=PEEK(I+489)
2720 2720 P491=PEEK(I+490):P492=PEEK(I+491)
2730 2730 P493=PEEK(I+492):P494=PEEK(I+493)
2740 2740 P495=PEEK(I+494):P496=PEEK(I+495)
2750 2750 P497=PEEK(I+496):P498=PEEK(I+497)
2760 2760 P499=PEEK(I+498):P500=PEEK(I+499)
2770 2770 P501=PEEK(I+500):P502=PEEK(I+501)
2780 2780 P503=PEEK(I+502):P504=PEEK(I+503)
2790 2790 P505=PEEK(I+504):P506=PEEK(I+505)
2800 2800 P507=PEEK(I+506):P508=PEEK(I+507)
2810 2810 P509=PEEK(I+508):P510=PEEK(I+509)
2820 2820 P511=PEEK(I+510):P512=PEEK(I+511)
2830 2830 P513=PEEK(I+512):P514=PEEK(I+513)
2840 2840 P515=PEEK(I+514):P516=PEEK(I+515)
2850 2850 P517=PEEK(I+516):P518=PEEK(I+517)
2860 2860 P519=PEEK(I+518):P520=PEEK(I+519)
2870 2870 P521=PEEK(I+520):P522=PEEK(I+521)
2880 2880 P523=PEEK(I+522):P524=PEEK(I+523)
2890 2890 P525=PEEK(I+524):P526=PEEK(I+525)
2900 2900 P527=PEEK(I+526):P528=PEEK(I+527)
2910 2910 P529=PEEK(I+528):P530=PEEK(I+529)
2920 2920 P531=PEEK(I+530):P532=PEEK(I+531)
2930 2930 P533=PEEK(I+532):P534=PEEK(I+533)
2940 2940 P535=PEEK(I+534):P536=PEEK(I+535)
2950 2950 P537=PEEK(I+536):P538=PEEK(I+537)
2960 2960 P539=PEEK(I+538):P540=PEEK(I+539)
2970 2970 P541=PEEK(I+540):P542=PEEK(I+541)
2980 2980 P543=PEEK(I+542):P544=PEEK(I+543)
2990 2990 P545=PEEK(I+544):P546=PEEK(I+545)
3000 3000 P547=PEEK(I+546):P548=PEEK(I+547)
3010 3010 P549=PEEK(I+548):P550=PEEK(I+549)
3020 3020 P551=PEEK(I+550):P552=PEEK(I+551)
3030 3030 P553=PEEK(I+552):P554=PEEK(I+553)
3040 3040 P555=PEEK(I+554):P556=PEEK(I+555)
3050 3050 P557=PEEK(I+556):P558=PEEK(I+557)
3060 30
```









# PAGE 6 THE MAGAZINE

FOR ALL ATARI<sup>®</sup> COMPUTER<sup>®</sup> OWNERS

\*ISSN 0950-9303/930001



THE BEST  
PROGRAM  
LISTINGS  
from  
U.S.A.  
U.K.  
AUSTRALIA  
PUBLIC  
DOMAIN  
SOFTWARE  
LIBRARY  
SPECIAL  
OFFERS



NEWS  
REVIEWS  
TUTORIALS  
UTILITIES  
HINTS &  
TIPS  
plus more

PAGE 6 is published bi-monthly.

Annual Subscription is £7.00. Send TODAY to:

**PAGE 6, P.O. BOX 54,  
STAFFORD, ST16 1DR**

Tel. 0785 41153

## CONTACT

- **PEN PAL** required: write to me and discuss our common interest in all items in Atari Ray Charles 150 Hurdles Great Countdown 3268. Virginia Australia.
- **PEN PAL** required: exchange information & self-correspondence with particular interest Atari User Groups to Michael Kröner, Chrysler 343, D 3060 Berlin 37 Germany.
- **PEN PALs** required for The Portland Atari Club in Oregon, please contact Margaret Mearns 6 Lee Road, apt 2552 N.E. 46 Avenue, Portland, Oregon 97212 U.S.A. if you would like to exchange disks and other information.
- **CHALLENGED Power Pac**: Have recently purchased one of these and I would like to buy some software for it or find some technical details on how the unit works. Is there anybody who could help me on this? Contact: Damaris, 16 Highbury Road, Cornwall, St Austell, Linn. Phone 0405-61383
- **HOFTINGHAM MICRO CLUB** meets at the Castle Gate Congressional Privatisation Building opposite Media 6, Spencer in Nottingham town centre at 7.30pm. The Atari section of the club meets on the 2nd and 4th Mondays of every month. All are welcome. Or write for more details to: Steve Hall, 26 Chapel Lane, Ravenhill, Nottingham NG10 8DA.
- **NEWSLETTER EXCHANGE** - Atari Hardware User Group P.O. Box 251, Coderham, New York 11316 U.S.A. Contact: Jonathan Novick - also can be reached at Madras Madras 3355 (24 hr) Sat 01699
- I have a Tektronic CP300K printer and I am experiencing difficulty with replacement ribbons, the last few I've bought were dry and were no good. Has anybody had similar problems and knows of a solution? Contact: Ian Fowell, 34 North Pole, Oakley, Beds MK43 7PS
- I am in the process of constructing a bulletin board phone number list for the Late Valley Atari Users Group. If anyone has any details regarding a bulletin board they have logged onto or a board they run, please send info on the board itself, operating times, board rules, special features, etc. All details will be merged into a searchable list by 1996. Contact: Matthew Tydeman, 125 Cadmore Lane, Clencham, Walsham, Woss, Herts SG8 6HL

## ATARI LOVERS SWEATSHIRTS & T-SHIRTS

### from only £4.50!

The new range of ATARI LOVERS Sweatshirts & T-shirts for men and girls by Churchill Cavendish.

Quality UK made garments emblazoned with the slogan  
**ATARI LOVERS HAVE MORE BYTE!**

Choice of SIX COLOURS & THREE SIZES -  
S (32"-36") M (36"-42") L (42"-48")

**T-SHIRTS £4.50** Crew neck short-sleeved T-shirts WHITE GREY RED

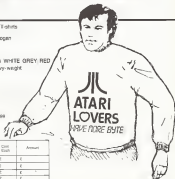
**SWEATSHIRTS £5.99** Top quality crew neck heavy-weight long sleeved sweatshirts  
MAROON GREEN ROYAL BLUE

#### ORDER FORM

Name  
Address

Send off today to Churchill Cavendish, Timothy Lane  
BATLEY, W. Yorks WF17 0BA. 01937-52714-52724  
Please allow up to 14 days for delivery  
P & P FREE on orders over £20. It would help if you  
could give an alternative colour choice in Column 5.

T-Shirt or Sweatshirt (Enter T or S)	No. of Pkgts	Size (S-M-L)	Colour Choice	An Other Colour	Cost Each	Amount
					£	£
					£	£
					£	£
					£	£
PLEASE ADD 1% PER-PACKAGE IF ORDER EXCEEDS 1000						£
TOTAL TWENTY POUNDS ONLY						£



## MONITOR ON DISK

Like the look of a program? You can find time to buy it in! You've asked the staff three times to do it for you while you're out at work, and the staff hasn't. Or maybe you have typed it in but it won't run. Then why not take all the effort out of it and send for the **MONITOR DISK**. All the main programs in each issue of **MONITOR** are now available just recorded on disk for you. They cost £4.95 which includes postage and packing, send a cheque/postal order made payable to the "U. K. Asia Computer Distributors Ltd to Monitor Magazine, P.O. Box 3, Rayleigh, Essex. If you live in Europe add 50p, if outside Europe add £1.50. Please allow 28 days for delivery.

### Monitor Disk 8

Includes: **Quarkplot**, a fast Graphics 8 Plot/Drawing package; **Nightmare Reflections**, an excitingly frustrating adventure; **Matchbox**, improve your concentration with this memory game; **Interupts**, 5 demo programs showing various uses of interrupts.

### Monitor Disk 9

Includes: **Keyo**, a new typing checker; **Mailboat** (Borland), database program for Multiboot disk; **Bincode**, binary code from Basic; **Happytyper**, automatic line numbering; **RanDisk**, for use with the 1300SE; **Fast Mail**, a speedy shape filling utility.

### Monitor Disk 10

Includes: **3D Maze**, escape from the maze in time if you can; **PCB Patterns**, destroy your enemies before they get you; **Disk Jacket**, useful programs for making your own disk covers; **Crash**, an excellent game, not to be missed.

### Monitor Disk 11

Includes: **Hexadecimal Code Generator**, better presentation for your programs; **Cracking the Code**, seven cilla programs from the series; **RAM Tester**, with a little bit of hardware and this program, you can hear your own voice. (for 400/600 only); **Bonus Programs**: **Home/FM**, a useful utility for use with Home/Filing Manager to give quick access to data disks.

## BACK ISSUES

Previous issues of the magazine are obtainable from the club for £3 plus 50p postage each. They contain many interesting and informative articles, hints & tips, program listings for you to copy, reviews and practical advice. If you have missed out send for your copies of back issues today! Please note that issues 1, 2, 3, 5, 7 are already sold out.

### Issue 4.

Includes a complete in-depth look at Display Lists, what they are, how to use them, LMS explained, horizontal and vertical scrolling, etc. Another article shows how to get text on a Graphics 8 screen and gives an example graph to prove the point. A comprehensive review of many of the different types of joystick that are available gives ratings for comfort, action, looks and value. Program listings are plentiful and include 'Pedemoni', a BASIC version of a well known arcade game; **Stunt Rider** in which you must jump your motorcycle over the buses; **Hex** is a two player board game with excellent graphics, and for the more serious minded, you can enjoy designing your own shapes with GAD (computer assisted design).

### Issue 5.

The first part of the series on 'Cracking the Code' starts in this issue and covers Binary, Hexadecimal and Decimal mathematics. There is an article on protecting your BASIC programs from prying eyes and an interesting article on hardware modifications to the 800/400 machines to give improved sound and picture quality, a cold start key and a busy light for your cassette player. Also included is a review of the new programming language 'Action' showing its potential for creating exciting fast action games. Games listings shown include **Gal-bert**, which is a 'Q-bert' type game; also **Droptier** in which the player must cross the drawbridge dodging the dragons firing breath at



reach the treasure room. Other listings include a label maker and a QMA locator for Radio Amateurs.

### Issue 6.

Includes a useful tutorial showing how to paint **Neopainter** and **Versa** screen pictures, also contains a terrific program demonstrating 80 characters across the screen. A new regular column for adventure enthusiasts is started to give reviews of adventure games and give hints and tips on how to play them. Part two of **Cracking the Code** continues with addressing modes and binary sums. The hardware design for a **Light Pen** is shown together with some sample programs use with it once you have built it. Fun with Art from **Egypt** is reviewed and some of the excellent results of using this package are shown. Programs

include **Planotron** and a **RTTY** listing for use with a short wave band radio; the **Atan 880** interface and a signal terminal unit (such as the Maplin TU1000).

### Issue 8.

Contains a preview of the new **Acad** computers. Two new series start: one about how this week and the other 'Starting from BASIC' for beginners. **Cracking the code** continues and concluding part of 'Interupts' discusses horizontal and vertical scrolling. The adventure column includes reviews of **Mask of the Sun** and **Sorcerer**. Other reviews include **Conan**, **Spy vs Spy**, **Alley Cat** and **Gloobusters**. Programs are **Matchbox**, a concentration game; **Quarkplot**, a Graphics 8 Plot/Drawing utility; and **Nightmare Reflections**, an excitingly frustrating adventure.

### Issue 9.

Includes a **RANDISK** for the 1300SE as well as a review of this excellent machine. Introduction to **MIDI**, just what is it? **ICETO** typing checker program. Utility to give binary code files from Basic. Reviews of **TopDOS**, **Homeview** and **My DO!**. Overview of **PORTIN** as an alternative to Basic. Utility to fill in shapes on Graphics 8 and last too! **Thriller** on Las Valley Area Club. **HAPPY TYPER** gives automatic line numbers and programmable function keys. Utility for indexing 'Matchbox' data.

### Issue 10.

Includes all the facts about **Digital** Patterns. **Disk Jacket**, a neat program for making your own disk covers. Opening Out: more about how this works. Reviews of the **Great American Road Race**, **Kennedy Approach**, **Red Moon**, **Asylum** and **Webbinger**. Two excellent games: **PCB Patterns** and **3D Maze**. Introduction to the world of communications. Continuation of regulars: **Cracking the Code**, **Starting from Basic** and **What's MIDI?**

THE CHART  
TOPPING FLIGHT  
SIMULATION  
NOW ON  
ATARI

FREE COLOUR  
POSTER AVAILABLE  
WITH DISK VERSION

# FIGHTER PILOT

ALSO AVAILABLE  
ON 48K SPECTRUM,  
CBM64 & AMSTRAD

BY D.K. MARSHALL  
ADAPTED BY NIK SEN

**ATARI 400/800/600 XL  
800 XL/130 XE  
WITH AT LEAST 48K.**



For SPECTRUM simulation  
it stands far ahead of the runway benchmark!

SPECTRUM VERSION VOTED  
"Simulation of the year" - 1987  
by CRASH MAGAZINE READERS

#### ATARI

Fighter Pilot Disk

4 or 8 inch Floppy

**SPECTRUM**

Combat Disk

4 or 8 inch Floppy

**COMMODORE 64**

Speed King Disk

4 or 8 inch Floppy

**AMSTRAD CPC 600, 640, 6128**

Fighter Pilot

Fighter Pilot Disk

☐ £12.95

☐ £8.95

☐ £8.95

☐ £8.95

☐ £7.95

☐ £12.95

☐ £8.95

☐ £12.95

☐ £8.95

☐ £12.95

☐ £8.95

☐ £12.95

Orders payable to Digital Integration Ltd

Enclosed is cheque/£ O.T. for \_\_\_\_\_ Total

Name \_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Please debit my Access/Visa card no

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Expiry date

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Please send to Digital Integration

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Washminster Trade Centre Washminster Road Canterbury Surrey QM11 9 5AJ

WAT and a. is a subsidiary of D.I.L. (Production) (UK) Limited

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Please send to

**DIGITAL  
INTEGRATION**



Washminster Trade Centre  
Washminster Road  
Canterbury Surrey  
QM11 9 5AJ



Trade and Export enquiries  
welcome Lifetime Guarantee

A NEW CLASSIC

FROM PAUL WOARIES  
AUTHOR OF ENCOUNTER

# YOU CHOOSE THE ACTION IN MERCENARY

Mercenary — a unique combination of flight simulation, adventure and arcade fun. You choose the action.

There is complete freedom of movement in a truly three-dimensional vector-graphic environment. Graphics of exceptional speed create a very realistic experience.

You'll never play the same game twice. Random elements attempt to seal your fate. Your interaction is crucial.

Mercenary presents an absorbing challenge that you will accept again and again.

## MERCENARY ESCAPE FROM ZARG

Fly above the planet Zarg in a variety of craft. Investigate the credibility of structures and the prime nation as you quest for extended supplies. Search the dimensional subterranean complex in pursuit of your fate.

Become entangled in the continuing conflict between the Vikings and Mechanoids. Accept their paid assignments or work independently to gain wealth.

You only have one bar friend — the ever reliable Element, an almost human PC, who guides you in your quest.

Pursue your objective as you will: On success you have to escape from Zarg.

INITIAL RELEASE

FOR COMMANDORE 64 AT 1988 £19.95 AND 1989 £19.95

CASSETTE £9.95

DISK £12.50

NOVAGEN SOFTWARE LTD'S SALES OFFICES: 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 442, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 476, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498, 500, 502, 504, 506, 508, 510, 512, 514, 516, 518, 520, 522, 524, 526, 528, 530, 532, 534, 536, 538, 540, 542, 544, 546, 548, 550, 552, 554, 556, 558, 560, 562, 564, 566, 568, 570, 572, 574, 576, 578, 580, 582, 584, 586, 588, 590, 592, 594, 596, 598, 600, 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624, 626, 628, 630, 632, 634, 636, 638, 640, 642, 644, 646, 648, 650, 652, 654, 656, 658, 660, 662, 664, 666, 668, 670, 672, 674, 676, 678, 680, 682, 684, 686, 688, 690, 692, 694, 696, 698, 700, 702, 704, 706, 708, 710, 712, 714, 716, 718, 720, 722, 724, 726, 728, 730, 732, 734, 736, 738, 740, 742, 744, 746, 748, 750, 752, 754, 756, 758, 760, 762, 764, 766, 768, 770, 772, 774, 776, 778, 780, 782, 784, 786, 788, 790, 792, 794, 796, 798, 800, 802, 804, 806, 808, 810, 812, 814, 816, 818, 820, 822, 824, 826, 828, 830, 832, 834, 836, 838, 840, 842, 844, 846, 848, 850, 852, 854, 856, 858, 860, 862, 864, 866, 868, 870, 872, 874, 876, 878, 880, 882, 884, 886, 888, 890, 892, 894, 896, 898, 900, 902, 904, 906, 908, 910, 912, 914, 916, 918, 920, 922, 924, 926, 928, 930, 932, 934, 936, 938, 940, 942, 944, 946, 948, 950, 952, 954, 956, 958, 960, 962, 964, 966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986, 988, 990, 992, 994, 996, 998, 1000

NOVAGEN